

International Technical Support Organization

The OS/2 Warp 4 CID Software Distribution Guide

January 1998



Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special Notices" on page 513.

First Edition (January 1998)

This edition applies to OS/2 Warp 4 in a software distribution environment and to NetView Distribution Manager/2 (NVDM/2) with Database 2 (DB2) Version 2.11 for OS/2 and Tivoli TME 10 Software Distribution 3.1.3 for OS/2 (SD4OS2) software distribution managers. This edition also applies to OS/2 Warp 4 subcomponents, such as TCP/IP 4.0, MPTS, NetFinity 4.0, Peer Services, and LAN Distance, and to OS/2- related products, such as eNetwork Personal Communications for OS/2 Warp, eNetwork Communications Server for OS/2 Warp, Transaction Server, Lotus Notes, and Lotus SmartSuite 96 for OS/2 Warp.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. DHHB Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 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

Figures	xiii
Tables	xvii
Preface	xix
How This Redbook is Organized	xix
The Team That Wrote This Redbook	xxi
Comments Welcome	xxiii
Redbooks Online	xxiii
<hr/>	
Part 1. Common Information	1
Chapter 1. Software Distribution Made Simple	3
1.1 Standard Installation Method	3
1.2 Installation by Replication	5
1.3 CID-Enabled Installation	6
1.4 CID Concepts and Terminology	8
1.4.1 Response Files	8
1.4.2 Redirected Installation (Redirected I/O or Drives)	10
1.4.3 Software Distribution Manager	11
1.4.4 Logging facilities	13
1.5 Installation Modes	13
1.5.1 Attended Installation	13
1.5.2 Lightly Attended Installation	14
1.5.3 Unattended Installation	15
1.5.4 Clarification	15
1.5.5 Summary	16
1.6 Code Server Setup	17
1.6.1 Server Types	17
1.6.2 CID Structure	17
Chapter 2. Introducing Software Distribution Managers	21
2.1 LAN CID Utility	21
2.1.1 LAPS/MPTS Versions	23
2.2 NetView Distribution Manager/2	24
2.3 TME 10 Software Distribution 3.1.3 for OS/2	25
2.4 Single-Server Software Distribution Network	27
2.5 Positioning LCU, NetView DM/2 and TME 10 SD 3.1.3 - Summary	33
2.6 Overview of Software Distribution Managers	34

Chapter 3. OS/2 Warp 4 - Installation Steps Local Install.	39
3.1 Installation Diskettes	39
3.2 Local Install Phase 1	40
3.3 Local Install Phase 2	44
3.4 Local Install Phase 3	53
3.5 The Installation Directory Structure	56
3.6 The Local Install Key Files	58
3.6.1 The INSTALL.EXE Executable (\OS2\INSTALL)	58
3.6.2 The CASAGENT.EXE Executable (\CID\LOCINSTU)	58
3.6.3 The LOCAL.COM Command File (\IBMINST\RSP\LOCAL)	58
3.6.4 The NPCONFIG.EXE Executable (\IBMINST)	59
3.6.5 The Response Files (\IBMINST\RSP\LOCAL)	59
3.7 FixPak Installation	61
3.7.1 The SERVICE Command	63
3.7.2 The FSERVICE Command	65
3.7.3 Sample ServicePak/FixPak Response File	67
3.7.4 Enhanced ServicePak/FixPak Response Files	67
3.7.5 Logging Information	70
3.7.6 Interrupted Service	71
3.7.7 CSF Response File Keywords	71
Chapter 4. Utilities to Ease Remote Installation.	75
4.1 Loading OS/2 CID Utilities to the Code Server	76
4.2 Unpacking the OS/2 Warp 4 CID Utilities	77
4.3 The GETOSCID Command.	78
4.4 The SEDISK Command	79
4.4.1 SEDISK Return Codes	80
4.5 The SEIMAGE Command.	81
4.5.1 SEIMAGE Return Codes.	82
4.6 The SEINST Command	82
4.6.1 SEINST and SEMAINT Return Codes.	84
4.6.2 RSPINST Return Codes	84
4.7 Display Driver Install	89
4.7.1 Matrox Millenium Driver Installation for IBM PC 350	92
4.8 Necessary Utilities for LAN CID Utility (LCU)	96
4.8.1 The GETBOOT Command	96
4.8.2 The GETREXX Command	96
4.9 The SrvIFS Utility	98
4.9.1 The THINSRV Command	99
4.9.2 The THINIFS Command	101
4.9.3 The IFSDEL Command.	106
4.10 LAN CID Utility (LCU)	106
4.10.1 The CASINSTL Command	107

4.10.2	The CASAGENT Command	110
4.10.3	The CASCKREX Command	111
4.10.4	The CASDELET Command	112
4.11	MPTS Utilities	112
4.11.1	The THINLAPS Command	112
4.12	NetView DM/2: The NVDMBDSK Utility	118
4.13	General CID Return Codes	120
4.14	Desktop Shuffler	121
4.14.1	Example of the SHUF_CTL.INI Control File	122
4.14.2	The Working Control and Keywords Files	125
4.15	Remote Multiple Printer Support	131
4.15.1	Remote Printer Install (RINSTPRN)	131
4.15.2	The RINSPRN Command	131
4.15.3	Remote Printer Install Response File Keywords	136
4.15.4	RINSTPRN Response File Configurator RINSTCFG	143
4.15.5	Backing Up and Restoring Printer and Job Properties	147
4.15.6	Holding or Releasing a Queue	149
4.16	Software Installer	150
4.16.1	Parameter Definitions	151
4.16.2	Product Database File	152
4.16.3	Installation Scenario	152
4.17	ASD - Alternative Software Delivery	154
4.17.1	ASD-Allowed and ASD-PreReq Service and Support Issues	156
4.17.2	ASD Requirements	157
4.17.3	Design Assertions	158
4.18	Feature Installer (FI)	159
4.18.1	Using the Command Line Interface (CLIFI)	159
Chapter 5. Loading Product Images to the Code Server		163
5.1	Loading OS/2 Warp 4 - Base Operating System	163
5.2	Loading OS/2 Warp 4 - Networking Components	163
5.2.1	MPTS Refresh Paks	163
5.2.2	Loading MPTS Utilities	164
5.3	Loading the Mini Server SrvIFS	165
5.4	Loading eNetwork Communications Server for OS/2 Warp	165
5.4.1	Communications Manager/2 Advanced Features	165
5.4.2	Personal Communications Lite	165
5.5	Loading eNetwork Personal Communications for OS/2 Warp	165
5.5.1	Copying from Diskettes	166
5.5.2	Copying from CD-ROM	166
5.6	Loading Database 2 for OS/2	166
5.6.1	Copying from Diskettes	166
5.6.2	Copying from CD-ROM	167

5.7	Loading OS/2 Warp Server File and Print Product Images	167
5.8	Loading CICS Images from CD-ROM	168
5.9	Loading MQSeries from CD-ROM.	168
5.10	Loading NetView DM/2 Diskette Images.	169
5.11	Loading Lotus Notes OS/2 Client Product Images	169
5.12	Loading Lotus SmartSuite 96 Product Images	170
Chapter 6. Remote Installation Commands for Additional Products.		171
6.1	eNetwork Communications Server for OS/2 Warp	171
6.2	eNetwork Personal Communications for OS/2 Warp.	173
6.2.1	Base Product Installation	174
6.2.2	Advanced Features Installation.	176
6.3	Database 2 for OS/2.	177
6.4	MQ Series for OS/2	177
6.5	Transaction Server Installation (CICS)	178
6.6	Lotus Notes Remote Installation.	179
6.7	Lotus SmartSuite 96 Remote Installation	180
Chapter 7. Creating Response Files		181
7.1	OS/2 Warp 4 - Base Operating System	182
7.1.1	Use of Client-Specific Response File for OS/2 Installation	184
7.1.2	Detailed Keyword Description.	185
7.1.3	New Keywords in OS/2 Warp 4.	204
7.1.4	Printer Description Table PRDESC.LST	209
7.1.5	Include Keywords Processing Sequence	211
7.1.6	The User Exit Keywords of the Response File	214
7.2	Feature Installer Response File	216
7.2.1	Example of an OS/2 Warp 4 Response File	217
7.3	MPTS Response File	219
7.3.1	Building a Configuration File.	220
7.3.2	Use of Client-Specific Response File for MPTS Installation	225
7.3.3	Example of an MPTS Response File.	226
7.4	OS/2 Peer Response File.	230
7.4.1	Building a OS/2 Peer Response File.	230
7.4.2	Example of an OS/2 Peer Response File	231
7.5	OS/2 LAN Requester Response File	233
7.5.1	Example of an OS/2 LAN Requester Response File	234
7.6	TCP/IP Response File	236
7.6.1	Example of a TCP/IP Response File.	239
7.7	LAN Distance Response File	240
7.7.1	LAN Distance Response File Keywords	240
7.7.2	Example of a LAN Distance Response File.	241
7.8	NetFinity Response File	241

7.9	Software Installer Response File	244
7.10	Network Signon Coordinator/2 Response File	245
7.11	SystemView Agent Response File	245
7.11.1	Specific Keywords	246
7.12	Mobile File Sync (MFS) Response File	247
7.13	Database 2 for OS/2 Response File	247
7.13.1	Use of Client-Specific Response File for Database 2 Installation	248
7.14	MQ Series Response File	250
7.15	Transaction Server (CICS) Response File	251
7.15.1	Server Response File	251
7.15.2	CICS Client Response File	252
7.16	Feature Install (CLIFI) Response File	254
7.16.1	Creating and Reading a Response File	254
7.16.2	Response File Example	255
7.16.3	FI Response File Keywords	258
7.17	Remarks on Netware Requester Installation	274
7.18	Communications Server and Advanced Features Response Files	274
7.18.1	Use of Client-Specific Response File	276
7.19	eNetwork Personal Communications Response File	277
7.20	Lotus Notes OS/2 Client Response File	282
7.21	Lotus SmartSuite 96 Response File	283

Part 2. Installing and Using Software Distribution 285

Chapter 8. Optimizing OS/2 Warp 4 Boot Diskettes	287
8.1 Creating OS/2 Boot Diskettes	287
8.1.1 Contents of Diskette 0 (OS/2 Warp 4 Installation Diskette	288
8.1.2 Contents of Diskette 1 (OS/2 Warp 4 Diskette 1)	288
8.2 Adding Network Adapter Support	293
8.2.1 NetBIOS Support	294
8.2.2 TCP/IP Support	295
8.2.3 Installing TCPBEUI Support	298
8.3 Adding a Redirector	301
8.3.1 ANXIFS	301
8.3.2 SrvIFS	303
8.3.3 NFS	303
8.4 Adding Software Distribution Agent	304
8.4.1 LCU	305
8.4.2 NVDM/2	306
8.4.3 TME10 SD 3.1.3	307

Chapter 9. LAN CID Utility (LCU) / SrvIFS	311
9.1 Hardware and Software Recommendations	311
9.1.1 Estimated Space Requirements of Product Images	312
9.2 Setting Up the LCU Code Server	313
9.2.1 Base Installation	313
9.2.2 Recommended CID Directory Structure	313
9.2.3 Copying REXX to the Code Server	314
9.2.4 Server-Enabling of the Code Server using SrvIFS	315
9.2.5 Other Information on the Code Server	316
9.3 Preparations at the Client Workstations	318
9.3.1 Requirements on the Clients.	318
9.3.2 Creating LCU Boot Diskettes	319
9.4 LCU Command File	322
9.4.1 General information on Command Files	322
9.4.2 Working with Default Response Files and LCU Command Files	324
9.4.3 LCU Command File Structure	326
9.4.4 BASE.CMD Command File	334
Chapter 10. NetView Distribution Manager/2 (NVDM/2)	339
10.1 NetView DM/2 Overview	339
10.1.1 Components of NetView DM/2	339
10.1.2 NetView DM/2 Workstation Types.	340
10.2 Prerequisites	341
10.2.1 NetBIOS Resource Considerations	341
10.2.2 Adding Network Adapters to MPTS.	342
10.2.3 Database 2 for OS/2 Considerations.	343
10.2.4 User Profile Management (UPM)	344
10.2.5 Modifying STARTUP.CMD Before Installing Netview DM/2	345
10.3 Installing and Configuring NetView DM/2	345
10.3.1 Installing NVDM/2 Server through a Response File	346
10.3.2 Installing NVDM/2 Server Interactively	347
10.4 NetView DM/2 Databases.	351
10.5 Global Names, Objects and Catalogs	352
10.5.1 Global Names.	353
10.5.2 Objects	353
10.5.3 Catalog	353
10.5.4 Change Files	353
10.6 Response files for NetView DM/2	354
10.7 Preparing a Client Workstation	354
10.7.1 Creation of Boot Diskettes	354
10.8 REXX Enablement	358
10.8.1 Setup at the CC Server.	359
10.9 Creating Change Files	361

10.9.1	The REXX Start Profile	361
10.9.2	The FDISK and Format Profile	361
10.9.3	The OS/2 Warp 4 Profile	362
10.9.4	The OS/2 Warp 4 Maintenance Partition Profile	363
10.9.5	The MPTS Install Profile	364
10.9.6	The NVDM/2 CC OS/2 Client Install Profile	366
10.9.7	The Feature Installer V1.2.1 (FISSETUP) Change Profile	370
10.9.8	The Command Line Interface Feature Installer Change Profile	371
10.9.9	OS/2 Warp 4 FixPak 4 Profile	372
10.9.10	Display Driver Install for S3 Step 1	373
10.9.11	Display Driver Install for S3 Step 2	374
10.9.12	Display Driver Install for S3 Step 3	375
10.9.13	TCP/IP Change Profile	375
10.9.14	Network SignOn Coordinator Change Profile	376
10.9.15	OS/2 Peer Services Change Profile	377
10.9.16	Personal Communications for OS2 Warp Profile	378
10.9.17	Remote Printer Install (RINSTPRN) Change Profile	379
10.9.18	Desktop Shuffler Profile	380
10.10	NetView DM/2 Command Overview	381
10.10.1	Creating the Software Catalog	383
10.11	Connectivity between CC Server and CC Client	385
10.11.1	Define CC Clients using CDM Commands	385
10.11.2	Define CC Clients using the Graphical User Interface	386
10.11.3	Establish Connectivity at the CC Client Workstation	387
10.11.4	Install Change Files on Client Workstations	388
10.11.5	Using the CDM Install Command	394
Chapter 11.	TME 10 Software Distribution 3.1.3 for OS/2	395
11.1	Environment	395
11.2	Installation Steps for OS/2 Distribution Server	396
11.3	Editing the Base Configuration File	402
11.4	Image/Directory Setup	403
11.4.1	Setup Activities	403
11.4.2	Additional Hints and Tips	408
11.5	Setup Activities	408
11.5.1	Prepare Response Files at SDSERV1	408
11.5.2	Configure Redirection Support	409
11.5.3	Additional Sample Profiles	411
11.5.4	Prepare Pristine Client Boot Diskettes	413
11.6	Create the Change Files	415
11.7	Install the Pristine Client	417
11.7.1	Define Client at SDSERV1	417
11.7.2	Define Client Specific Installation Parameters	417

11.7.3	Submit Installation Commands at SDSERV1	418
11.7.4	Boot CLIENT1 from Diskettes	418
11.7.5	Monitor Installation Results	419
11.7.6	Verify Request Completion	420
11.8	OS/2 Pristine Installation over TCP/IP	421
11.8.1	Prepare Response Files at SDSERV1	422
11.8.2	Configure Redirection Support	424
11.8.3	Prepare Pristine Client Boot Diskettes (OS/2 Warp 4)	425
11.8.4	Creating Change Files	429
11.9	Install the Pristine Client	432
11.9.1	Define Client at SDSERV1	432
11.9.2	Define Shared Installation Parameters	432
11.9.3	Define Client Specific Installation Parameters	433
11.9.4	Submit Installation Commands at SDSERV1	433
11.9.5	Boot CLIENT1 from Diskettes	434
11.9.6	Monitor Installation Results	435
11.9.7	Verify Request Completion	436

Part 3. Appendixes 439

Appendix A. More Syntax Information and REXX Listings	441
A.1 The SrvIFS Utility	441
A.1.1 The SERVICE.INI Keywords	441
A.1.2 Example Authorization List File for SERVICE.EXE	446
A.1.3 Example SERVICE.INI File	447
A.2 Autopartitioning the Hard Disk	448
A.2.1 The Fixed Disk Utility Program (FDISK)	449
A.2.2 Functionality of FDISK	450
A.2.3 FDISK Command Line Interface	452
A.2.4 FDISK ReturnCodes	456
A.2.5 Remarks	457
A.2.6 The FDISK /FILE Parameter	457
A.3 SETBOOT	458
A.4 Partitioning Utility: DISKPRP	460
A.5 The Desktop Shuffler Tool	463
A.5.1 CLNDESK Syntax	463
A.5.2 Return Codes	464
A.5.3 Control File Syntax	464
A.5.4 Errors and Error Recovery	469
A.5.5 Limitations and Resitrictions	469
A.5.6 The SHUF_CTL.INI Control File	470
A.5.7 The SHUF_KEY.INI File	475
A.6 TME 10 SD 3.1.3 Base Configuration Parameter	475

A.6.1 Sample Base Configuration File	480
Appendix B. More Hints and Tips for OS/2 Warp 4 Installers	481
B.1 Using RSPDDI for Device Driver Installation	481
B.2 Installing OS/2 Printer FixPaks Using RINSTPRN	482
B.3 ABIOS Problems on RAM-Loadable ABIOS Systems	484
B.4 FDISK Install Notes	485
B.4.1 IDE Hard Drive BIOS Setup	485
B.4.2 FDISK Return Code Information	486
B.4.3 FDISK Command Line Examples	487
B.5 OS/2 Warp 4 CD-ROM Installation Notes	488
B.5.1 Updating SNOOP.LST and .SNP Files	490
B.5.2 Adding and Changing Device Drivers	492
B.5.3 System Hang During Bootup	492
B.5.4 Using RESERVE.SYS to Solve a Problem	492
B.6 Creating OS/2 Warp 4 NFS CID Boot Diskettes	492
B.6.1 Other MPTS CID Considerations When Using NFS	497
B.6.2 LCU Considerations	501
B.6.3 Known MPTS Problems and Workarounds	501
Appendix C. Latest News on Netscape 2.02 Refresh and Java 1.1.4	503
C.1 Installing Java 1.1.4 Unattendedly	503
C.1.1 Creating the Java 1.1.4 Response File	503
C.1.2 Starting the Unattended Installation	506
C.1.3 Performing an Unattended Uninstall	506
C.2 Installing Netscape 2.02 Refresh Unattendedly	507
C.2.1 Creating the Netscape Response File	507
C.2.2 Initiating Remote Install	510
C.3 Installing Netscape 2.02 OS/2 Plugins Unattendedly	510
C.3.1 Initiating Remote Install	512
Appendix D. Special Notices	513
Appendix E. Related Publications	517
E.1 International Technical Support Organization Publications	517
E.2 Redbooks on CD-ROMs	517
E.3 Other Publications	518
How To Get ITSO Redbooks	519
How IBM Employees Can Get ITSO Redbooks	519
How Customers Can Get ITSO Redbooks	520
IBM Redbook Order Form	521

List of Abbreviations	523
Index	527
ITSO Redbook Evaluation	531

Figures

1. International Project Members	xxii
2. Example of an OS/2 Response File (Extract)	10
3. Recommended CID Directory Structure	18
4. MPTS, LCU, SRVIFS Constellation	22
5. Single Server Software Distribution Network	28
6. Interconnected Domains	29
7. Software Distribution from MVS	31
8. Software Distribution from AIX	32
9. Complex Distribution Network through SNA/APPC Connection	35
10. Software Distribution through Push Mode	36
11. Software Distribution through Pull Mode	37
12. Local Install Phase 1	43
13. Local Install Phase 2	45
14. The System Configuration Screen	51
15. The Network Component Installation Screen	52
16. Local Install Phase 3	54
17. OS/2 Warp 4 Directory Structure	57
18. Serviceable Products on The System	64
19. The Files are Archived or Backed Up before Servicing	64
20. Base Service Response File	67
21. Service Response File to Support First-Time-Invocation	68
22. Service Response File to Support Existing Archive	68
23. Service Response File to Support Backing Out an Archive	69
24. Service Response File to Support Backing Out to a Backup	69
25. Service Response File to Support Committing	70
26. Service Response File to Support Archive Redirection	70
27. Loading Essential OS/2 Warp 4 CID Utilities	77
28. Loading Required DLLs for Remote Printer Install	77
29. Necessary DLL for HPFS Partitions	78
30. .DSC Files and the Corresponding Chip Sets (Part 1 of 2)	91
31. .DSC Files and the Corresponding Chip Sets (Part 2 of 2)	92
32. The Matrox Millennium Configuration Program	95
33. Statements added to CONFIG.SYS by THINIFS	105
34. CONFIG.SYS File after CASINSTL	110
35. PROTOCOL.INI Created by THINLAPS	116
36. CONFIG.SYS after THINLAPS Command	117
37. NVDMBDSK - Boot Diskette Update Window	119
38. RMPI Configurator Window	144
39. Print Driver - Properties Window	144
40. Print Device - Properties Window	145

41. Local Print Queue - Properties Window	146
42. OS/2 Software Choice Web Site as of 12/97	154
43. Install Object - Inventory Window	156
44. CMSETUP Program Invocation out of LCU	173
45. PCOMOS2 Install Program Invocation out of LCU	176
46. CM/2 Advanced Features Install out of LCU	176
47. Extract of an LCU Client Command File for DB/2 Install	177
48. Extract of an LCU Client Command File for MQ Series Install	178
49. Extract of an LCU Client Command File for CICS Server Install	179
50. Extract of an LCU Client Command File for CICS Client Install.	179
51. Printer Description List PRDESC.LST	210
52. Include Example	211
53. IncludeInLine Example.	212
54. Embedded Include Keywords Example	214
55. Feature Installer Base Response File FIBASE.RSP (Extract).	217
56. Example of an OS/2 Response File (Part 1 of 2).	218
57. Example of an OS/2 Response File (Part 2 of 2).	219
58. Example of an MPTS Response File (Part 1 of 3).	227
59. Example of an MPTS Response File (Part 2 of 3).	228
60. Example of an MPTS Response File (Part 3 of 3).	229
61. Example of an OS/2 Peer Services Response File (Part 1 of 2)	232
62. Example of an OS/2 Peer Services Response File (Part 2 of 2)	233
63. Example of an OS/2 LAN Requester Response File (Part 1 of 2)	235
64. Example of an OS/2 LAN Requester Response File (Part 2 of 2)	236
65. Example of a TCP/IP Response File	240
66. Example of a LAN Distance Response File	241
67. Database 2 Client Application Enabler Response File	250
68. Example of a MQ Series Response File	251
69. Example of a CICS Client Response File	254
70. Lotus Notes Client Response File	283
71. Lotus SmartSuite 96 Response File	284
72. Example of a SERVICE.INI File.	315
73. Example of a SERVICE.INI File with Limited Client Access Security . . .	316
74. Corrective Service Diskettes Directory Tree	317
75. NVDM/2 Server Response File Used by the NVDMINST Command . . .	346
76. NVMD/2 Installation Screen.	347
77. Startup of Installation.	347
78. NetView DM/2 Base and Server Features Installation Window.	348
79. NVDM/2 Base and Server Features Installation (Configure) Window . . .	349
80. NVDM/2 Base and Server Features Installation - Directories Window . .	350
81. NetView DM/2 Installation Progress Window.	351
82. DB2 Database Director	352
83. NVDM/2 CONFIG.SYS Modifications	358

84. REXX Start Change Profile	361
85. FDISK Change Profile	362
86. OS/2 Warp 4 Change Profile	363
87. OS/2 Warp 4 Maintenance Partition (SEMAIN) Profile	364
88. MPTS Maintenance Environment Change Profile	365
89. MPTS Production Environment Change Profile	366
90. NVDM/2 Client Change Profile Part of a Corequisite Group Install	367
91. NVDM/2 Change Profile: Maintenance Partition Agent	368
92. NVDM/2 Change Profile: Maintenance Partition Agent (Alternative)	369
93. REXX Procedure that Copies the Second Agent (SD_MAINT.COMD)	370
94. FISETUP Feature Installer Change Profile	371
95. CLIFI Change Profile	372
96. OS/2 Warp 4 FixPak 4 (XR_M004) Change Profile	373
97. Display Driver Install (DSPINSTL) Step 1 Change Profile	374
98. Display Driver Install (DSPINSTL) Step 2 Change Profile	374
99. Display Driver Install (VCFGCID) Step 3 Change Profile	375
100. TCP/IP Change Profile	376
101. Network SignOn Coordinator Change Profile	377
102. OS/2 Peer Change Profile	378
103. Personal Communications for OS/2 Change Profile	379
104. Remote Printer Install Change Profile	380
105. Desktop Icon Shuffler Change Profile (SHUFFLER.PRO)	381
106. Desktop Icon Shuffler REXX Command File	381
107. Creating the NetView DM/2 Software Catalog	383
108. NetView DM/2 - CDM Catalog Window	384
109. Build Change File Window	384
110. NetView DM/2 - CDM CC Domain Window	386
111. New Workstation Window	387
112. Selecting Software in the CDM Catalog Window	390
113. CDM Catalog Install Window	390
114. Install Window	391
115. Install Order Window	392
116. Options Window: Corequisite Group	393
117. TME 10 SD 3.1.3 NetBIOS Implementation Environment	395
118. TME 10 Software Distribution 3.1.3 Installation Welcome Screen	397
119. Install Window	397
120. Install - Directories	398
121. Install Progress Window	398
122. Distribution Configuration - General Window	399
123. Distribution Window	400
124. Enterprise Connection Window	401
125. 'Save' Window	401
126. TME 10 SD 3.1.3 Directory Structure	403

127.Create Base Subdirectories	404
128.TME 10 SD 3.1.3 ANXSRV.INI File.....	410
129.TME 10 SD 3.1.3 ANXCLI.INI File	411
130.TME 10 SD 3.1.3 XR_M004.PRO Change Profile	412
131.TME10 SD 3.1.3 CLIFI.PRO Change Profile	412
132.TME10 SD 3.1.3 RMPI.PRO Change Profile	413
133.TME10 SD 3.1.3 SHUFFLER.PRO Change Profile	413
134.OS/2 Warp 4 Pristine Installation Profile WARP4.PRO.....	416
135.MPTS Pristine Install Profile MPTSNB.PRO (With NETBIOS)	416
136.ANXIFS Pristine Install Profile ANXIFS.PROI.....	416
137.SD4OS2 Client Install Profile SD4OS2NB.PRO	417
138.(STARTUP.MOD) - SD4OS2 Installation Modification File.....	420
139.NFS Response File NFS.RSP.....	423
140.OS/2 Warp 4 Profile (SD4OS2) for Pristine Install: OS2.PRO	430
141.MPTS-TCP/IP Profile (SD4OS2): MPTSIP.PRO.....	430
142.NFS Profile (SD4OS2): NFSPRIS.PRO	431
143.SD4OS2 Client Profile: SDPRISIP.PRO.....	431
144.SD4OS2 Pristine Client Install over TCP/IP: SD4OS2IP.PRO	431
145.STARTIP.CMD Returned Messages When Run Successfully	435
146.Example of an SrvIFS Authorization List.....	447
147.Example of SERVICE.INI	448
148.FDISKPM WIndow	450
149.Secondary CLIFI Response File CID.RSP to Install Java 1.1.4	504
150.Response File for Unattended Installation of Netscape	507
151.Response File for Unattended Installation of Netscape Plug-Ins	510

Tables

1. Basic Characteristics of the Standard Installation Method	5
2. Basic Characteristics of the Cloning Method	6
3. Characteristics of the Rapid Deployment Tools Replication Technique	6
4. Basic Characteristics of CID-Enabled Installations	7
5. Installation Modes (Summary)	16
6. Positioning LCU, NVDM/2, and TME 10 SD 3.1.3 (Part 1 of 2)	33
7. Positioning LCU, NVDM/2, and TME 10 SD 3.1.3 (Part 2 of 2)	34
8. The CONFIG.SYS Environment Variables in Phase 1(Part 1 of 2)	41
9. The CONFIG.SYS Environment Variables in Phase 1(Part 2 of 2)	42
10. The CONFIG.SYS Environment Variables in Phase 2 (Part 1 of 3)	48
11. The CONFIG.SYS Environment Variables in Phase 2 (Part 2 of 3)	49
12. The CONFIG.SYS Environment Variables in Phase 2 (Part 3 of 3)	50
13. Installation Programs (Part 1 of 2)	55
14. Installation Programs (Part 2 of 2)	56
15. Response Files Used in the Local Installation	59
16. TCP/IP Response File Keywords (Part 1 of 2)	238
17. TCP/IP Response File Keywords (Part 2 of 2)	239
18. Disk Space Recommendations for Product Images	312
19. Product Variable Description	328
20. NetBIOS Resources for NetView DM/2	342
21. Keywords for FDISK /QUERY (Part 1 of 2)	453
22. Keywords for FDISK/QUERY (Part 2 of 2)	454
23. Some FDISK Return Codes	456
24. Base Configuration File Parameters (Part 1 of 4)	476
25. Base Configuration File Parameters (Part 2 of 4)	477
26. Base Configuration File Parameters (Part 3 of 4)	478
27. Base Configuration File Parameters(Part 4 of 4)	479

Preface

This redbook describes the CID (Configuration, Installation and Distribution) enablement of OS/2 Warp 4 with all its subcomponents. It is also a handbook that provides step-by-step guidance in all phases of the usage and administration of the OS/2 base operating system, such as SEINST and the CLIFI (Command Line Interface Feature Installer, introduced with OS/2 Warp 4 for the first time), its subcomponents and main OS/2 products for the implementation of CID in an OS/2 LAN environment using LCU (LAN CID Utility), NetView DM/2 or TME 10 SD 3.1.3.

This document is intended for workstation specialists and system technical personnel responsible for mass distribution of OS/2 products in an OS/2 Warp 4 LAN. Some knowledge of LAN redirection principles and TCP/IP is assumed. This redbook solely focuses on OS/2 Warp 4. If CID-related information is needed for previous OS/2 versions, such as OS/2 Warp Connect or even OS/2 V2.11, or about software distribution techniques other than those mentioned above, you need to obtain the redbook titled *OS/2 Installation Techniques: The CID Guide*, SG24-4295.

This redbook gives a broad understanding of a new features introduced with OS/2 Warp 4 that were not available with previous OS/2 versions. It is essential to know about these new features, such as Feature Installer (FI), to successfully make use of them and distribute OS/2 Warp 4.

How This Redbook is Organized

This redbook is divided into three parts:

- Part I provides common information applicable to all CID issues.
- Part II provides software distribution-specific information regarding the platform you are using or going to use:
 - LAN CID Utility (LCU)
 - NetView Distribution Manager/2 (NVDM/2)
 - TME 10 Software Distribution 3.1.3 for OS/2 (SD4OS2)
- Part III, Appendixes, provides more detailed information about certain CID-related utilities and lists sample REXX files used in different software distribution scenarios.

The individual chapters are organized as follows:

- Part I
 - Chapter 1, “Software Distribution Made Simple” on page 3, provides basic CID-related information, such as CID concepts, installation modes and setup of code server information.
 - Chapter 2, “Introducing Software Distribution Managers” on page 21, introduces the three software distribution managers used in this redbook.
 - Chapter 3, “OS/2 Warp 4 - Installation Steps Local Install” on page 39, discusses the steps of a local install of OS/2 Warp 4 to give the administrator in charge of software distribution an idea of what is going on in a local install that can and must be used in a remote install.
 - Chapter 4, “Utilities to Ease Remote Installation” on page 75, discusses utilities that are used in software distribution scenarios and their return codes. This chapter also introduces Software Installer, Feature Installer and Alternative Software Delivery (ASD), which is a mechanism used by IBM’s Software Choice on the World Wide Web.
 - Chapter 5, “Loading Product Images to the Code Server” on page 163, illustrates the first part of a code server setup: Loading product images on the code server.
 - Chapter 6, “Remote Installation Commands for Additional Products” on page 171, illustrates remote installation commands for products other than OS/2 Warp 4 and OS/2 Warp 4-related products.
 - Chapter 7, “Creating Response Files” on page 181, discusses in detail how product-specific response files are created. All displayed response files can be used by all software distribution managers.
- Part II
 - Chapter 8, “Optimizing OS/2 Warp 4 Boot Diskettes” on page 287, shows methods how to optimize boot diskettes that are used to connect to a software distribution manager. It shows how to get the maximum out of the set of three OS/2 Warp 4 boot diskettes.
 - Chapter 9, “LAN CID Utility (LCU) / SrvIFS” on page 311, illustrates the use of LAN CID Utility (LCU) as the software distribution manager of choice.
 - Chapter 10, “NetView Distribution Manager/2 (NVDM/2)” on page 339, illustrates the use of NetView DM/2 as software distribution manager of choice. All information found here regarding software change files can be easily modified to be used by TME 10 SD 3.1.3. This chapter is the most detailed one concerning a superior software distribution manager. solution

- Chapter 11, “TME 10 Software Distribution 3.1.3 for OS/2” on page 395, illustrates the use of Tivoli’s TME 10 SD 3.1.3 as software distribution manager of choice. It also shows how to distribute software packages through TCP/IP and NFS.
- Part III
 - Appendix A, “More Syntax Information and REXX Listings” on page 441, provides more syntax information to certain CID-related utilities and lists a couple of REXX file used in software distribution scenarios.
 - Appendix B, “More Hints and Tips for OS/2 Warp 4 Installers” on page 481, provides information from PSP support people in Austin.
 - Appendix C, “Latest News on Netscape 2.02 Refresh and Java 1.1.4” on page 503, details how to install Java 1.1.4, Netscape 2.02 Refresh, and Netscape Plug-ins remotely.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Uwe Zimmermann is an Advisory Systems Engineer at the International Technical Support Organization, Austin Center. He has been working in heterogeneous networks for more than ten years. Before joining the ITSO, he worked in an IBM branch office in Stuttgart as a Networking Systems Engineer and was in charge of large account customers for more than five years. His areas of expertise include OS/2 LAN/Warp Server, Windows NT Server and Workstation, NetWare, Software Distribution (CID), dynamic TCP/IP, and Network Computing (NC).

Claude Chainier is a Systems Engineer in IBM Global Services in France. He has 28 years of experience in IBM. His areas of expertise include OS/2 Warp 4 and OS/2-related software (LAN Server, Communications Server, MQ Series, and so on).

Les Devine is an IT Specialist in IBM Global Services in South Africa. He has two years of experience in OS/2, NVDM/2 and OS/2-related software. He has worked at IBM for more than ten years. His areas of expertise include NVDM/2, NVDM/MVS and SWIM/2 (Software Installation and Maintenance).

Daniela Hübner is a Service Specialist with SerCon GmbH, an IBM Global Services company in Germany. She has three years of experiences in Images and Workflow, OS/2, Networking, several programming languages, and remote software distribution using the CID architecture.

Anssi Nummi is a Services specialist with IBM Finland. He has four years of experience in PC operating systems, mainly in support and maintenance but in education and software development as well. He has worked for IBM for more than two years, first in the OS/2 Support Group and later in the IBM Desktop Services/Global Services. His areas of expertise include workstation and server operating systems on the PC platform including OS/2, Windows NT and Windows 95.



Figure 1. International Project Members

Thanks to the following people for their invaluable contributions to this project:

Mel Bryant, IBM Austin, OS/2 Warp Server Customer Focal Point

Stephen Cunningham, IBM Austin, OS/2 and OS/2 LAN/Warp Server Support

Patrick Davis, IBM Austin, OS/2 Print Development

Steven King, IBM Austin, Product Manager OS/2 Warp and NC Server

Tom Lendacky, IBM Austin, Netscape Install

Patrick Nogay, IBM Austin, OS/2 Print and Spool Development

International Technical Support Organization, Austin Center:

Marcus Brewer, Staff Editor

Tara Campbell, Technical Editor

Jürgen Friedrichs, Advisory Software Engineer

Alex Gregor, Senior Systems Engineer

Contributors at other IBM sites and at other companies:

Tom Arbuckle, formerly with IBM Austin, ITSO Austin Center

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in Appendix , "ITSO Redbook Evaluation" on page 531 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web Sites:

For Internet users <http://www.redbooks.ibm.com>

For IBM intranet users <http://w3.itso.ibm.com>

- Send us a note at the following address:

redbook@vnet.ibm.com

Redbooks Online

The page "Redbooks Online!", is **the** source for finding complete redbooks on the World Wide Web. You can view the complete contents of our books. The books are arranged within the structure of the IBM Redbooks CD-ROM libraries. Clearly, we would like to see a CD-ROM or hardcopy book as part of your daily reference material. These online books help you understand what is on our CD-ROMs, what is available on hardcopy, and give you immediate access to the latest books that are not yet distributed on plastic or paper media.

Redbooks Online is on the World Wide Web at the following Web site:

<http://www.redbooks.ibm.com>

Part 1. Common Information

2 The OS/2 Warp 4 CID Software Distribution Guide

Chapter 1. Software Distribution Made Simple

As the number of LAN-attached workstations in your company continues to grow, so do the resources required to install and maintain software on those workstations. There is quite an easy way to handle remote software installation and keep costs down! It is called CID, which stands for Configuration, Installation and Distribution.

The CID architecture's primary goals are to:

- Simplify the installation of software on pristine OS/2 workstations
- Reduce the time required to perform software installations
- Centralize the configuration and remote installation of software
- Reduce software installation costs
- Eliminate human intervention at the target workstation when preparing and executing the configuration, installation, migration, and maintenance processes that are necessary to operate the workstation and maintain the software on it
- Enable the code executed at the target workstation to perform all required configuration and installation tasks, including the integration of previous customization
- Centralize human intervention at a central preparation site

Software that is capable of being distributed and configured through the LAN is called CID-enabled. CID-enabled products can be configured and installed remotely on LAN-attached clients with limited or no interaction required locally at each client. Although powerful in concept, implementing the processes involved with the CID architecture can be quite complex.

In order to evaluate this software distribution and installation process, a brief explanation of the characteristics of the two most common software installation methods is provided below.

1.1 Standard Installation Method

The most common method of installing workstation software is to install from diskettes or a CD-ROM. This method includes the following critical factors:

- Human intervention is required to customize the product by passing configuration information to the installation program through its dialog interface. This process must be performed by a person who is familiar with

this dialog interface, with the product features to be installed to meet the end user's needs, and with the system environment where the product is installed.

- Since most of today's products are shipped on large numbers of diskettes or CD-ROMs, media exchange is required during the installation process.
- Information about the progress of the installation process as well as information about whether the process completed successfully must be checked to guarantee a fully operational system.
- Some software requires the workstation to be rebooted in order to activate configuration changes.

The last three tasks also require human intervention. Although they do not require as much system-specific knowledge as the first task, they may require some basic knowledge about how to install workstation software. In order to achieve the goal of unattended installation, all of these tasks must be executed automatically.

With the standard installation method, an installation program is shipped with the product and is tailored to the specific installation needs of that product. With this installation program, one critical factor is automated: the installation program contains logic to check underlying hardware and software to determine which code modules need to be installed on the workstation and which files (such as CONFIG.SYS or *.INI) need to be updated.

The standard installation method requires the end user to provide installation and configuration information at the time of product installation. Thus, product configuration and product installation are not separable tasks. Installation and configuration information must be provided at the same time by the same person at the workstation itself. In addition, skilled people must be present at the workstation to perform this standard installation process. In other words, this installation method is not feasible if the installation process must be managed remotely.

Table 1 briefly summarizes the basic characteristics of the standard installation method:

Table 1. Basic Characteristics of the Standard Installation Method

Ability to exploit the software product's tailored installation program at the target workstation.	X
Ability to remotely manage the process of software configuration, installation, migration, and maintenance. No human intervention at the target workstation is required.	–
Ability to migrate previous customization.	X

1.2 Installation by Replication

To overcome the drawback of not being able to manage a standard installation remotely, a technique known as *replication* is used. Replication executes an installation procedure previously written by an administrator. This procedure is built by performing the following steps:

1. Install the product at the preparation site in the same way in which it would be customized and installed at the target workstation by using the installation program delivered with the product.
2. If the installation program provided with the product is not fully automatic, and requires user input, create a remote installation procedure that is not dialog-driven. This is done by recording the steps that are performed by the installation program (for example, which files have been installed, and which existing files have been updated with what kind of data). This can be a laborious and error-prone process. In this process, it is often necessary to discover steps that were previously intentionally hidden.
3. Invoke the software distribution manager at the target workstation. This can be done either by a local administrator or by a program. By using a program, the process does not require human intervention since the customization dialog is eliminated.

The replication method may not be suited to installing software on target workstations that have a different hardware or software configuration than the preparation site workstation.

Although replication achieves the goal of minimizing human intervention at the target workstation by centralizing installation tasks at the administration site, it requires reverse engineering the installation process, and distinguishing and copying the configuration differences between the administrator and the target workstation.

In addition, replication does not migrate software from previous customization unless you use the new tools developed by IBM's PSP division. This set of tools called *Rapid Deployment Tools* make use of replication and configuration techniques to allow both installation and migration of OS/2 workstations. The tools are made available only through the Rapid Deployment Team within NCSD (Network Computing Software Division). Contact person is Ken White. He can be reached through his Internet user ID: whiteken@us.ibm.com

Table 2 briefly summarizes the basic characteristics of the replication method up to date:

Table 2. Basic Characteristics of the Cloning Method

Ability to exploit the software product's tailored installation program at the target workstation.	–
Ability to remotely manage the process of software configuration, installation, migration, and maintenance. Human intervention at the target workstation is commonly not required although some operating systems still have human intervention dependencies.	X
Ability to migrate previous customization.	–

Table 3 briefly summarizes the characteristics of the replication method through Rapid Deployment Tools (Version-to-Version Tools and Enterprise Rescue Tools):

Table 3. Characteristics of the Rapid Deployment Tools Replication Technique

Ability to exploit the software product's tailored installation program at the target workstation.	X
Ability to remotely manage the process of software configuration, installation, migration and maintenance. Human intervention at the target workstation is commonly not required.	X
Ability to migrate previous customization.	X

1.3 CID-Enabled Installation

In order to combine the strengths of both the standard and the replication installation methods, the CID-enabled installation has been developed with the basic objective of performing remote unattended installation of software. It implements both the use of the original installation program of the product at the target workstation and the capability to invoke and manage the installation process from a central site.

The benefits of the CID-enabled installation method are:

- The specific installation program of the product can be exploited for both locally-managed as well as remotely-managed installation processes. The logic of the installation program can be used to discover the underlying hardware and software to determine which code modules are to be installed and which files need to be updated.
- Specifying installation and configuration information can be done at a different time and a different place than installing the software. This allows the preparation and installation processes to be divided into two individually executable tasks.
- Information that an installation program normally requests the user to provide at installation time can be specified by a central administrator at preparation time. This information is recorded in a response file. During product installation, this response file is used to provide the installation program with installation and configuration information.
- In order to reduce the labor of loading diskettes, the product's code images can be transferred from the original medium to a medium that is large enough to store the entire code image. This preparation step is performed before the installation process is started. During the installation process, these code images are accessed.
- A facility is provided to control the installation process remotely and to check whether the installation was completed successfully. If required, the workstation is automatically restarted.

The benefits listed above eliminate human intervention at the target workstation and leave any remaining manual tasks to central administrators. Thus, end users do not need to be involved in any of the preparation or installation tasks. In addition, the strengths of the product-specific installation program may be retained and exploited.

Table 4 briefly summarizes the basic characteristics of CID-enabled installations:

Table 4. Basic Characteristics of CID-Enabled Installations

Ability to exploit the software product's tailored installation program at the target workstation.	X
Ability to remotely manage the process of software configuration, installation, migration, and maintenance. No human intervention at the target workstation is required.	X
Ability to migrate previous customization.	X

It is the purpose of this document to detail software distribution and installation processes that do not require any human intervention at the target workstation and that make use of the product's installation program. Therefore, the CID-enabled installation method is used in the scenarios in this publication.

1.4 CID Concepts and Terminology

There have been many independently developed solutions which have been designed to take advantage of a LAN-based file server to distribute OS/2 operating system files, and of course, application programs and data files to various clients. However, many of these solutions operate on the replicating principle. For example, the LAN Download Utility (LDU), which is a feature of IBM NetView Distribution Manager/2 Version 2.1 uses a replication technique.

In the past, the replicating approach to operating system installation was one of the few ways to successfully install multiple systems simultaneously. Beginning with the introduction of OS/2 V2.0, a new approach to installation has been taken to provide support through the installation program itself. The two primary enhancements to the installation process brought about by CID are:

- Response File Support - The capability to provide predefined responses to any prompts normally aimed at the user during the installation or configuration process. This allows user interaction with the installation process to be bypassed.
- Redirected Installation - The capability to install from a drive other than A:. This drive could be an alternate drive on the target system, a redirected drive on a LAN or other network, or some other device that appears to the operating system as a logical drive (such as a CD-ROM device).
- Software Distribution Manager - The ability to control the process of installation as well as configuration for several products within one process. This gives the advantage of a better control of the overall installation process.

1.4.1 Response Files

Response files are product-specific ASCII files that contain sequences of keyword-value pairs. They are interpreted during the installation and configuration process of a product by the installation (and configuration) program. The keywords used in a response file are usually unique to each product.

Response files may include keywords that are specific to either the installation process or the configuration process or both.

Installation keywords provide the capability to predefine the responses to any prompt that the user would encounter during a standard product install. Therefore, with a properly prepared response file, a CID-enabled subsystem or application may be installed without requiring a user to respond to prompts during the installation process.

Configuration-related keywords may also be used during the installation in order that both installation and configuration occur concurrently. Configuration keywords may also be used after an installation to modify or reconfigure a currently installed system.

The example shown in Figure 2 shows you the **keyword=value** relationship for the disk-formatting option and display selection as part of the response file for the OS/2 CID installation procedure.

```

...
*****
* FormatPartition *
* Specifies whether or not to format the install partition*
* Valid Params: *
* 0=Do not format (DEFAULT) *
* 1=Format *
*****
FormatPartition=1
...

*****
* DisplayAdapter *
* Specifies which adapter should override the primary *
* adapter detected by the install process *
* Valid Params: *
* 0=Accept as correct (DEFAULT) *
* 1=Other than following (DDINSTAL will handle) *
* 2=Color Graphics Adapter *
* 3=Enhanced Graphics Adapter *
* 4=Video Graphics Adapter *
* 5=8514/A Adapter *
* 6=XGA Adapter *
* 7=SVGA Adapter *
*****
DisplayAdapter=4
...

```

Figure 2. Example of an OS/2 Response File (Extract)

The response file example shown above displays keywords and values for the disk-formatting option and displays selection of an OS/2 installation.

1.4.2 Redirected Installation (Redirected I/O or Drives)

A regular product installation is started from drive A: by inserting the installation diskette into drive A: and starting the installation program from the command line by entering the name of the installation program. It will continue to install from drive A: until all diskettes required to install the product have been fed into the diskette drive A:.

Redirected I/O defines the capability of installation programs to use a drive other than the A: drive, especially drive letters that are not connected to local drives (neither logical nor physical) but to drives, directories or subdirectories on a remote workstation.

Throughout this book, the workstation that uses a remote (redirected) drive is known as the client or redirector, and the workstation that provides a remote (redirected) drive is known as the server, software distribution server, or code server.

The client workstations will access the drive on the server where the product images reside and will perform the installation.

Depending on the method of communications used, there are different ways to connect to a code server and obtain a redirected drive, such as:

- SrvIFS Utility
- TCP/IP - Network File System (NFS)
- Novell NetWare
- OS/2 Warp Server and Remote Initial Program Load (RIPL)
- NetView DM/2
- TME 10 SD 3.1.3

In most cases, the redirected drive will be accessed through a Local Area Network (LAN). We will make this assumption throughout this document.

All descriptions in this book are based on IBM token-ring technology, NetBIOS or TCP/IP protocols. If your network utilizes other LAN hardware, corresponding drivers are needed to establish hardware connections and low-level protocols.

1.4.3 Software Distribution Manager

It is important for later sections that we now define the functions of server and client systems in a CID environment. In a CID environment, a code server is the system that contains the source files (or installation diskette images) to be used during the installation or maintenance process. The source files (or installation diskette images) for each product to be installed need to be placed onto the server in a predefined, specific format and structure. The specifics of this structure are unique to the individual products, and talking about diskette images means copying the contents of the product diskettes to the server into a structure using the diskette volume labels as subdirectory names. In most cases, utilities are provided with the application to ensure that the files from the product diskettes are properly transferred to the code server.

Aside from containing the files and programs required for installation, in some environments the server may also initiate and/or manage the installation of

code in one or more of its clients. In this case, the code server provides more function than just file sharing. For the purposes of this document, we will call such a system a software distribution manager and abbreviate it as SDM.

Recall that, in standard installation method, the person installing a particular software product manually invokes the product's installation program and provides it with installation and configuration information as well as product code that is usually stored on diskettes. This person also controls the progress of the installation process and may need to reboot the workstation after the installation program successfully terminates its execution.

These are the basic tasks that are to be performed by a software distribution manager. Aside from the basic idea of using programs to automate possible tasks, this approach has two major impacts:

- Human intervention at the target workstation is no longer required (thus, end users do not have to be involved in the installation process and skills can therefore be concentrated in central administrators).
- Difficult or repetitive tasks may be automated.

The functions and features of the software distribution manager (SDM) determine which, if not all, of the CID tasks may be automated. Some software distribution managers, such as NetView DM/2 V2.1, implement, for example, functions to schedule or remotely invoke software installation processes. Others, such as LAN CID Utility (LCU), do not have a scheduling capability.

The features of the particular software distribution manager also determine within which system environments it is able to drive the automated installation process. Additionally, these features decide whether this process is required to be invoked locally (at the target workstation) or whether it may be invoked remotely (at the client or server) or at the central site.

A system that is being installed, configured or maintained, is called the client. It utilizes the resources of a code server to gain access to the files and programs it requires and in some cases will operate under the direction of an SDM.

If software is installed by a computer-based software distribution manager, this SDM must provide some means to enable cooperation between both the client and the server system. Thus, a software distribution manager typically consists of a client component and a server component.

1.4.4 Logging facilities

A CID-enabled product defines log files that will be used to store information about the progress of product installation, configuration or maintenance. These files will contain information such as:

- Installation/maintenance/configuration update history
- Error information

Log information is stored in ASCII format. Again, using only command line parameters, the administrator must be able to define the drive and path where these files will be written.

During installation, some of the CID-enabled products also display progress indication information on the screen of the target workstation. This is in addition to what is written to the log files and contains a subset of the log file information

1.5 Installation Modes

Although this document uses the term installation, we should point out that this includes migration and maintenance as well. Unless otherwise noted, making the discussed products CID enabled includes installation, migration and maintenance. The installation techniques used to install any kind of software product are classified into three modes:

- Attended
- Lightly attended
- Unattended

1.5.1 Attended Installation

Attended installation is defined as that requiring a knowledgeable individual to be in attendance at the workstation where the software is being installed. This individual will typically need to respond to the various prompts that are displayed during the installation and configuration process.

1. Diskette-Based Installation

The standard installation method, which is diskette-based or diskette and CD-ROM based, is a prime example of the attended mode of installation. In this environment, the user is responsible for:

- Initiating the installation
- Responding to all installation and configuration prompts

- Inserting diskettes as appropriate
- Handling any errors that may occur
- Rebooting the system if required

2. Redirected Installation

With the capability of a redirected install, attended installation can also be performed without diskettes by accessing the diskette images through an alternate drive such as a local redirected drive or a redirected drive on a LAN. This type of installation is very similar to the diskette-based installation that users are familiar with today. Users will have to be present during the software installation to answer any questions asked by the product installation program.

The difference(s) between the redirected and the diskette-based installation is that:

- You don't have to carry all required diskettes and/or CD-ROM around to the end user for installation.
- You don't have to maintain all of these sets of diskettes and/or CD-ROM
- You don't need to have a machine with a CD-ROM unit if software is distributed on such a medium.

All of these points represent significant savings.

It should also be noted that the installation will typically proceed faster because the installation process will not have to pause while the user removes and replaces diskettes as prompted.

The actual drive used can be any medium which can be represented by a disk drive letter.

1.5.2 Lightly Attended Installation

The phrase *lightly attended installation* refers to an environment where an individual must be present to initiate the installation process and potentially perform other simple or predefined tasks. However, this individual would require no specialized system knowledge. The tasks that this individual would need to perform would typically be limited to:

- Starting the process
- Removing/replacing diskettes when prompted
- Rebooting the system if required

In some cases, an administrator may choose to require the individual attending the installation to provide predefined responses to the installation process.

In a CID environment, the tasks would most typically be performing a three diskette boot sequence or issuing a simple command to initiate the install process. In most cases, this kind of installation will be driven by a response file. Once the process is initialized, it should proceed to conclusion with no further interaction required by the individual.

1.5.2.1 Pristine Installation Boot Diskettes Content

If pristine installation, boot diskettes must be used. They have to contain all the necessary files to start and initiate the CID process. Typically, they will contain

- Necessary files to boot OS/2
- MPTS files to support LAN adapter with the right protocol
- Files to initiate a requester to remote drives
- Start-up file to initiate the process

1.5.3 Unattended Installation

Unattended installation has no requirement for an end user or administrator to be present at the system being installed. This is the most complex scenario. In this instance, the invocation of the install is handled by a software distribution manager (SDM).

To allow the software distribution manager to dictate when the installation should begin and what should be installed, the client system must have a distribution agent installed. This agent is started when the client system is IPLed. When a software distribution manager wishes to start installing a product, it will communicate with an agent that will prepare itself to execute the install process defined by the software distribution manager.

1.5.4 Clarification

In addition to the descriptions above, further clarification of lightly attended and unattended may prove useful. These terms are used throughout this document and the product documentation and may appear to have conflicting associations.

In an environment with no software distribution manager, product installation is always attended, though it may be lightly attended. A user must initiate the

installation process even though it may require little or no interaction from the user after it has been started.

In an environment with a software distribution manager, the individual product installation process should run in an unattended mode. However, the entire process (which includes the initiation of the software distribution manager and its client components), may indeed require a user at the workstation in lightly attended mode.

Therefore, when discussing unattended versus lightly attended, one must keep in perspective whether the discussion is aimed at an individual product installation process or the larger system installation process.

1.5.5 Summary

Table 5 summarizes the characteristics of the installation modes:

Table 5. Installation Modes (Summary)

Installation Modes	Diskettes		Redirected Drives	
	Dialog-Driven Installation and Configuration	Response File	Dialog-Driven Installation and Configuration	Response File
Attended	Standard installation User initiates, feed diskettes, provides responses	n/a	CID redirection only User initiates, provides responses	n/a
Lightly attended	n/a	CID response file User initiates, feeds diskettes	n/a	CID response file and procedures User initiates
Unattended	n/a	n/a	n/a	CID response files and procedures No end user required SDM required
Note: n/a: not applicable				

The CID concept only deals with slightly attended and unattended installations.

1.6 Code Server Setup

This section illustrates briefly the set up of a code server. After having decided what type of software distribution manager you will be using, the main steps are the same independent of the server or software distribution manager type. Before you start installing the server and products to be distributed, make sure you have adequate hardware available to be used for the server. File transfers, such as CID installations, consume quite a lot of CPU cycles. The more workstations being installed remotely, the faster the hard drive and the faster the processor.

1.6.1 Server Types

Depending on what software distribution manager and what network transport mechanism you will be using, there are several types of servers supported to hold the data structure requested for CID. LAN CID Utility has the ability to redirect drives located on other servers or even on host disks, like Novell NetWare servers or TCP/IP servers with NFS installed or Client Access/400 and AS/400.

Basically, all software distribution managers introduced in this redbook support any kind of file servers. For example, if you want to hold the CID directory tree on an OS/2 Warp Server (which perfectly makes sense), install either requester or peer server software on your software distribution manager machine to log on to OS/2 Warp Server to access the CID directory tree through an alias or share name.

1.6.2 CID Structure

One of the first steps of setting up the code server is to create a subdirectory structure on the server to put the image on, store the response files, have a place for the log files, and so on. Your CID directory can look like the one displayed in Figure 3 on page 18.

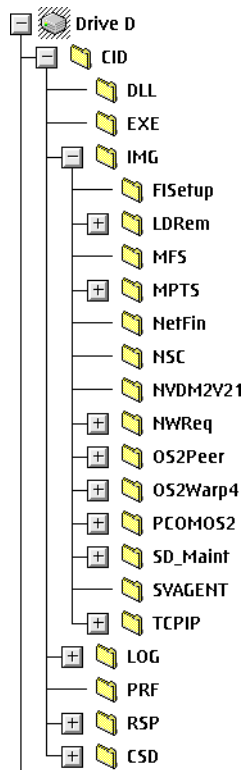


Figure 3. Recommended CID Directory Structure

To ensure that everything goes where it is supposed to, you might decide to set up the server manually. The first step is to create the CID structure mentioned before. For each product you will be installing using CID, you need to have a subdirectory underneath the IMG directory for the installation files and one for the response files underlying the RSP directory. Both directories are read-only to the client to make sure nothing is destroyed when working at the client workstation using the redirected drive on the server. You also have to create a subdirectory within the LOG directory, which has read/write access for the client workstation.

In case of LAN CID Utility, you will need a subdirectory, which will be called \CID\CLIENT, to hold the control files for a comprehensive installation process of each workstation.

The \CID\PRF directory will hold profile change files for NetView DM/2 and/or TME 10 SD 3.1.3. For FixPaks and ServicePaks, we recommend creating the \CID\CSD directory and underneath it the product-specific FixPak/ServicePak

name. For example, if you plan to distribute OS/2 Warp 4 FixPak 4 the directory to be created is \CID\CSD\XR_M004. Product refreshes, such as MPTS, should go directly into the product images directory (\CID\IMG\MPTS) rather than in the \CID\CSD tree.

In addition to the preparation of the products on the server, you will have to install all necessary client/server support files for the connection and redirection between server and client. And last but not least, you have to create start-up diskettes to start the pristine clients with.

Chapter 2. Introducing Software Distribution Managers

This chapter provides brief information about IBM's current workstation-based software distribution managers, such as:

- LAN CID Utility (LCU)
- NetView Distribution Manager/2 (NVDM/2)
- Tivoli TME 10 SD 3.1.3 (SD4OS2)

All three software distribution managers are being used in this redbook and have their own chapters.

2.1 LAN CID Utility

As far as networking and software distribution is concerned, Multiple Protocol Transport Services (MPTS) consists of three primary components:

- LAN Adapter and Protocol Support (LAPS)
- LAN CID Utility (LCU) and Code Server Setup Utility (CASSETUP)
- SrvIFS (Server Installable File System)

Although these components are packaged together, each component provides a separate function in the CID environment as shown in Figure 4 on page 22.

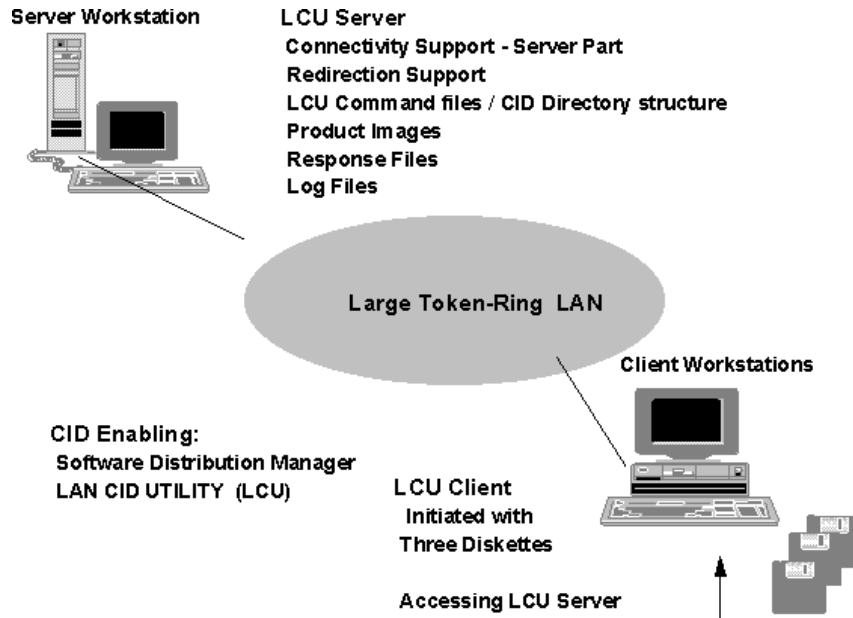


Figure 4. MPTS, LCU, SRVIFS Constellation

For example, OS/2 Warp Server 4.0 and the MPTS ServicePak WR08415 ship with six MPTS diskettes. Diskettes 1 to 4 provide network support and will be being prompted when installing MPTS. Diskette 5 provides all kinds of utilities, for example, LAN CID Utility (\LCU\LCU.ZIP) and Coder Server Setup Utility (\APPLETS\CASSETUP.ZIP). Diskette 6 provides MPTS-related publications in INF format.

1. LAN Adapter and Protocol Support (LAPS)

The LAPS component provides the LAN transport (network communication) subsystem for OS/2 environments. It is comprised of:

- NDIS-compliant protocol drivers
- NDIS-compliant network adapter drivers
- Support for Novell NetWare (IPX) and TCP/IP protocols
- OS/2 and DOS support for LAPS APIs (NetBIOS and IEEE 802.2).

2. LAN CID Utility

Integrated with MPTS is the LAN CID (Configuration Installation Distribution) Utility (LCU). This utility is designed to allow an administrator to chain together a series of CID installs. For example, an end user may require the OS/2, MPTS, DB2 for OS/2, and requester services of LAN

Server V4.0 to be installed. Although each product is individually enabled for CID, there is an obvious requirement to allow the complete install to be invoked as a single process instead of several individual processes. LCU provides this capability.

3. SrvIFS (Server Installable File System)

SrvIFS is actually a small NetBIOS-based file server and requester (THINIFS). Packaged with MPTS, this utility provides file redirection in a CID environment, therefore enabling clients to access code servers and consequently to install from diskette images.

The client function is initiated from a set of OS/2 boot diskettes. It cannot be initiated from a remote system.

SrvIFS is not a generalized LAN redirection product. It does not provide the equivalent functionality found in LAN products such as LAN Server V5.0.

2.1.1 LAPS/MPTS Versions

The first versions of LAPS were available as a separate product or packaged with other products like Extended Services V1.0 or IBM LAN Server V2.0. These early versions were not CID-enabled. The first CID-enabled versions of LAPS were shipped with IBM LAN Server V3.0 or with the Network Transport Services/2 (NTS/2) product, which was a subset (including LAN CID Utility or LCU) of LAN Server V3.0. LAPS and NTS/2 have been withdrawn from marketing as separate products. The successor of these products is MPTS, of which LAPS is a component, which features more functions and performance enhancements. Be aware that there are different flavors of MPTS.

Prior to OS/2 Warp Connect, the TCP/IP product provided a protocol stack to support TCP/IP applications, and MPTS provided socket support. Beginning with OS/2 Warp Connect, MPTS became the sole delivery vehicle for the TCP/IP protocol stack, merging with it the socket support. Therefore, TCP/IP Version 3.0 requires MPTS, which supplies the protocol stack required by TCP/IP. Thus, the two flavors are:

1. MPTS with an Unconverged Stack, as in OS/2 LAN Server V4.0
2. MPTS with a Converged Stack, as in OS/2 Warp Connect, OS/2 Warp Server and OS/2 Warp 4.

Why is it important to know about this difference? If you plan to update your MPTS with a CSD (Corrective Service Diskettes), make sure that you use the right one, as they require different CSD packages!

- Up to now, all MPTS versions have been downward compatible. So it is a good idea to always use the newest version. In our lab, we used MPTS V5.10, which comes with OS/2 Warp V4, and the MPTS product refresh diskettes WR08415.
- If you are installing several products that come with an MPTS of their own, their MPTS levels will likely be different because the products are developed and shipped on different schedules. Always install the most recent version, and never allow an installation program to downlevel your currently installed MPTS! MPTS installation checks to make sure you are always installing the latest level, and prompts you if you are not.

2.2 NetView Distribution Manager/2

Starting with Version 2 of NetView DM/2, the support of the first version of the product for distributing applications and data to programmable workstations in different network configurations was enhanced. NetView DM/2 V2.1 is available as three separate packages: the Entry package, the Extended package, and, as an additional feature, the Remote Administrator package, allowing customers to select the packages that meet their requirements.

Please see the chapter on NetView DM/2 packaging in the *IBM NetView Distribution Manager/2 Version 2.1 Installation and Customization Guide*, SH19-6915, for detailed information on how the product is bundled.

- The Entry package improves the support for host-connected OS/2 workstations while maintaining the Local Area Network (LAN) level of functions within the previous version of the product, including CDM (Change Distribution Manager), and LDU (LAN Download Utility).
- The Extended package includes the functions of the Entry package while providing substantial enhancements for the support of LAN-attached OS/2 workstations, with or without a host connection through its client/server capabilities.

The Remote Administrator feature is available with NetView DM/2 V2.1 as an additional feature of the Extended Package. It provides facilities to manage NetView DM servers (NetView DM/2 V2.1, NetView DM for NetWare and NetView DM/6000) in remote LANs that are APPC connected. The local NetView DM/2 server will become a manager of software distribution managers, thus providing a subset of the functionality offered by NetView DM on MVS.

In this publication, the term NetView DM/2 always refers to NetView DM/2 V2.1.

NetView DM/2 provides an SNA/DS (Systems Network Architecture/Distributed Services) transport function as well as software distribution management functions to exploit the CID installation process. Its client/server function operates across IBM OS/2 NetBIOS. NetView DM/2 itself is CID enabled.

To perform change management for the whole enterprise, NetView Distribution Manager for MVS should be used as the host. NetView DM/2 V2.1 is supported by NetView DM/MVS Release 5 (V1.5) or Release 6 (V1.6). It is a good idea to check the required APAR (Authorized Program Analysis Report) level before installing any of these products.

NetView DM/2 V2.1, in conjunction with NetView DM/MVS, provides a solution for an SNA network. It is also a key product for software distribution and remote change control in stand-alone LAN and interconnected LAN environments. NetView DM/2 V2.1 Extended package provides change management functionality and awareness down to the LAN client level that makes it an ideal product for managing OS/2 workstations in LAN networks.

The Remote Administrator feature offers an enterprise wide solution without necessarily using NetView DM on an MVS host.

2.3 TME 10 Software Distribution 3.1.3 for OS/2

TME 10 Software Distribution is Tivoli/IBM's solution for electronic software distribution in the enterprise network. It forms part of Tivoli Management Environment (TME) 10, Tivoli/IBM's solution for enterprise systems management.

For the examples described in this book, we used TME 10 Software Distribution for OS/2 V3.1.3, inherited from the former Tivoli TME and IBM SystemView families of products along with the following products:

- Tivoli/Courier 3.0
- Software Distribution for AIX, NT 3.1.3
- Netview Distribution Manager for AIX 1.2
- Netview Distribution Manager/2 2.1

This section describes how Software Distribution, a client/server systems management product, can:

- Electronically distribute and install software from a central site.

When a system software component or an application that runs on numerous workstations located at different sites has to be updated or upgraded, Software Distribution can completely automate the procedure. You can prepare and package software at one workstation, send it to the affected target workstations, and install it automatically.

You can facilitate the administration of software installation and maintenance on large numbers of heterogeneous workstations by taking advantage of Software Distribution's dynamic functions. By organizing your target workstations into dynamic groups whose members change according to the criteria you establish, you can selectively administer software. You can also organize the content of software packages dynamically in order to install only certain files on certain workstations.

- Keep track of hardware and software installed across the network on server and client workstations.

Software Distribution automatically stores a history and inventory record of all the hardware and software installed on each workstation in a network. This means that you are always aware of the configuration of all the workstations in your network and, consequently, of the activities that need to be performed to ensure consistency where you require it.

- Distribute and collect data.

System data files and user data files (such as user flat files or database exports) can be exchanged between the central site and target workstations and across workstations in the network. You can distribute data from the central site to targets by grouping workstations that need to receive the same files. Data distribution operations also provide data compression and page code translation options.

- Manage a multiplatform environment.
- You can take advantage of the benefits of Software Distribution in a multiplatform environment.

Workstations running OS/2, Windows 3.1, Windows 95, or Windows NT can be controlled from a Software Distribution server. Software Distribution servers can also interoperate with servers running different operating systems (such as AIX and Windows NT) in the same network. A NetView Distribution Manager for MVS Release 6 or Software Distribution for AIX site can also act as a central site for software and data distribution.

Software Distribution can, therefore, become a key element in ensuring the productivity and efficiency of your workstations and users by providing you with the means to:

- Save resources, time and money

When you install software manually on individual workstations, the process is time-consuming for both those who perform the installation and those who work at the workstation. When you use Software Distribution to automate software management, one person (the administrator) from one workstation can update thousands of computers, and plan the installation for a convenient time when it does not interfere with anyone's work schedule. What's more, you can automatically create backups of old software, which can be immediately reinstalled should an installation be unsuccessful.

- Improve efficiency

Controlling all the software installed on all the workstations means ensuring constant compatibility and consistency across your network.

You can use Software Distribution functions in many different network topologies, which differ greatly in complexity. In the simplest networks, Software Distribution is installed on a workstation that is referred to as the network distribution server. The other workstations in the network become distribution clients that can work in conjunction with a Software Distribution server.

2.4 Single-Server Software Distribution Network

A Single-Server Software Distribution network is a simple network with a server and its clients. Clients controlled by a server are referred to as local targets. As with Netview DM/2, the set of local clients, combined with the server that controls them, is known as a *change control domain*.

Figure 5 on page 28 shows an example of a software distribution network with a single software distribution server.

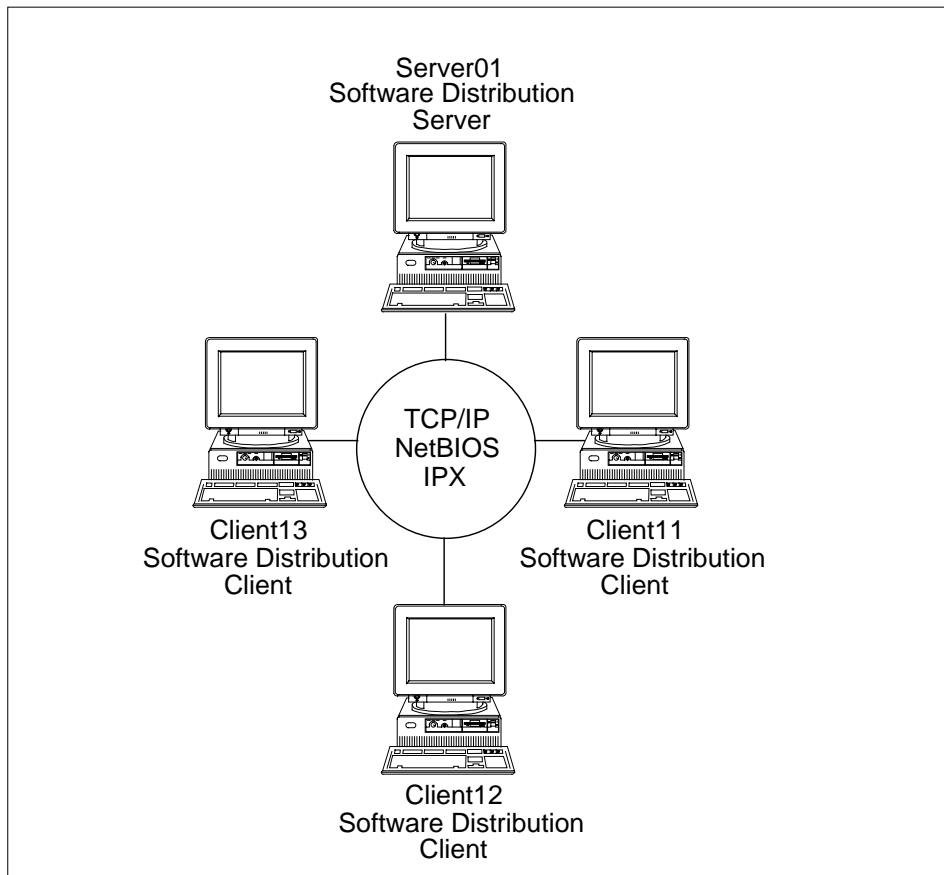


Figure 5. Single Server Software Distribution Network

Change control and distribution activity in a network is usually initiated from a server. However, if a client is configured with the necessary authorizations, it can be an active client, meaning that it can initiate operations on other clients in the network. A passive client can only have operations performed on it.

More complex networks can combine interconnected domains. Servers and clients in other domains are referred to as *remote targets*. Interconnected Domains are networks with two domains.

For example, as shown in Figure 6 on page 29, Server01 can perform data distribution, but not change control, on the clients in DOMAIN02.

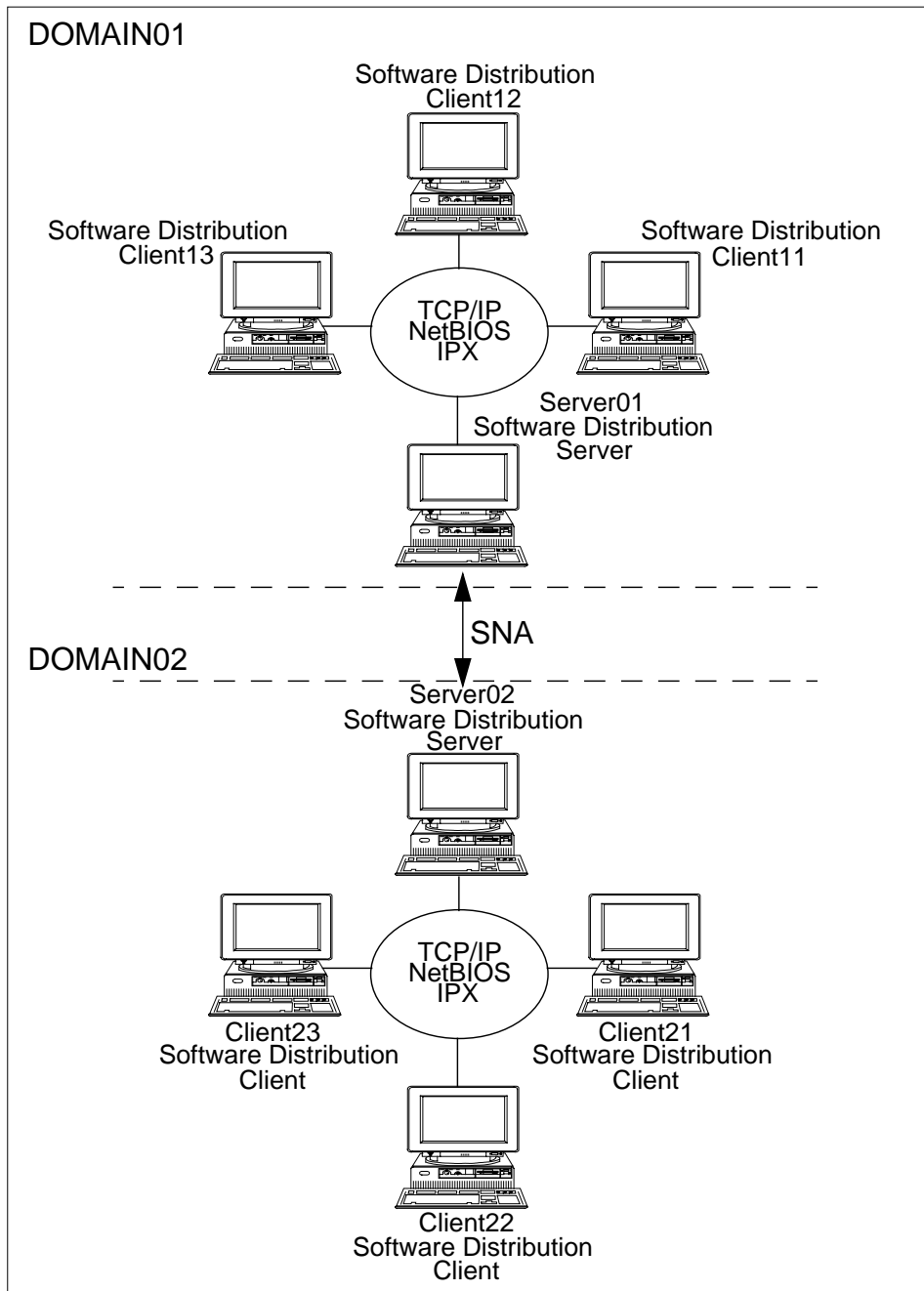


Figure 6. Interconnected Domains

Servers can be controlled from one or more central NetView Distribution Manager for MVS or Software Distribution for AIX sites that maintain a total picture of the entire change control network. Software Distribution from MVS shows a NetView Distribution Manager for MVS system managing multiple Software Distribution domains, and Software Distribution from AIX shows a similar configuration with a Software Distribution for AIX system as the central site.

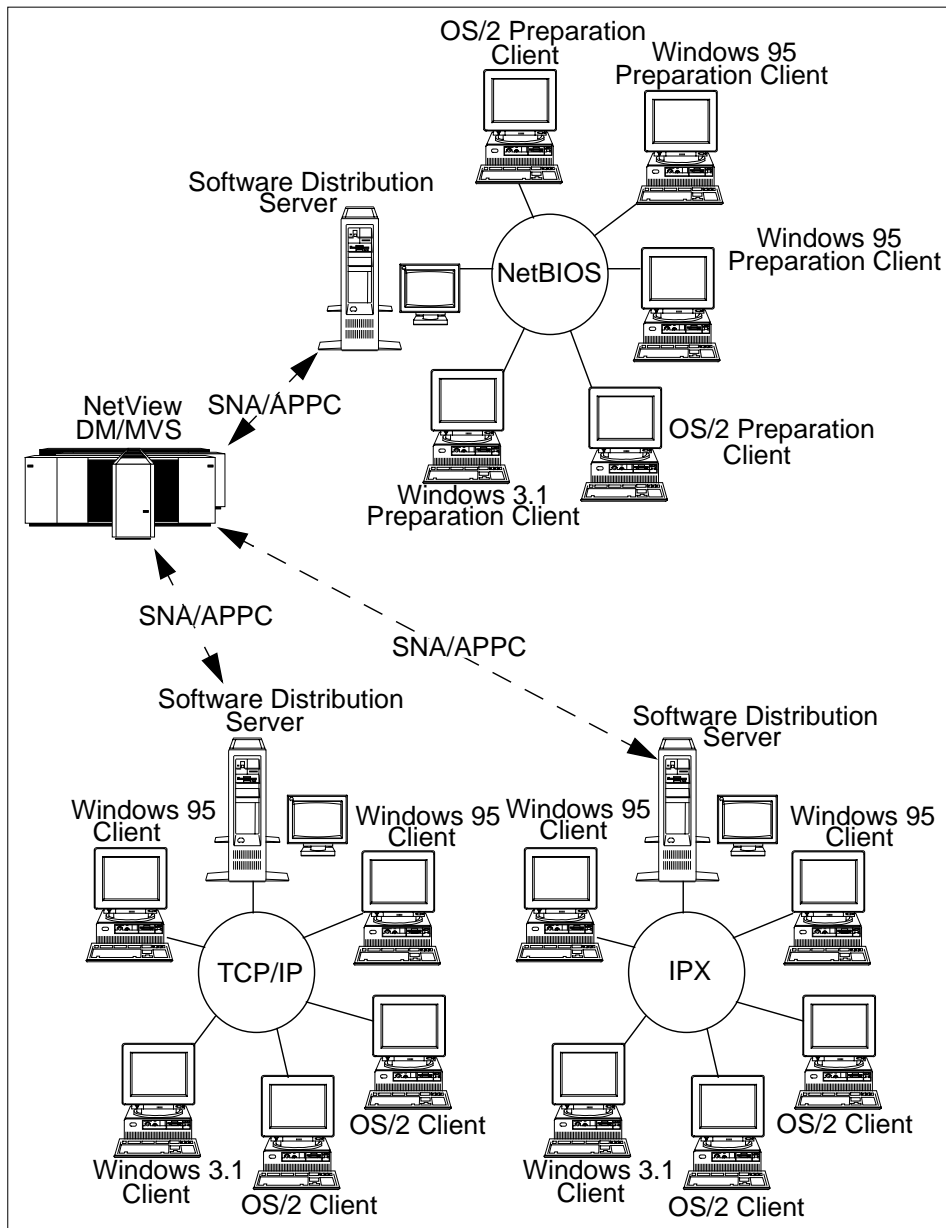


Figure 7. Software Distribution from MVS

Figure 8 on page 32 shows SD servers connected through the TCP/IP protocol, but serving clients through either TCP/IP, NetBIOS or IPX.

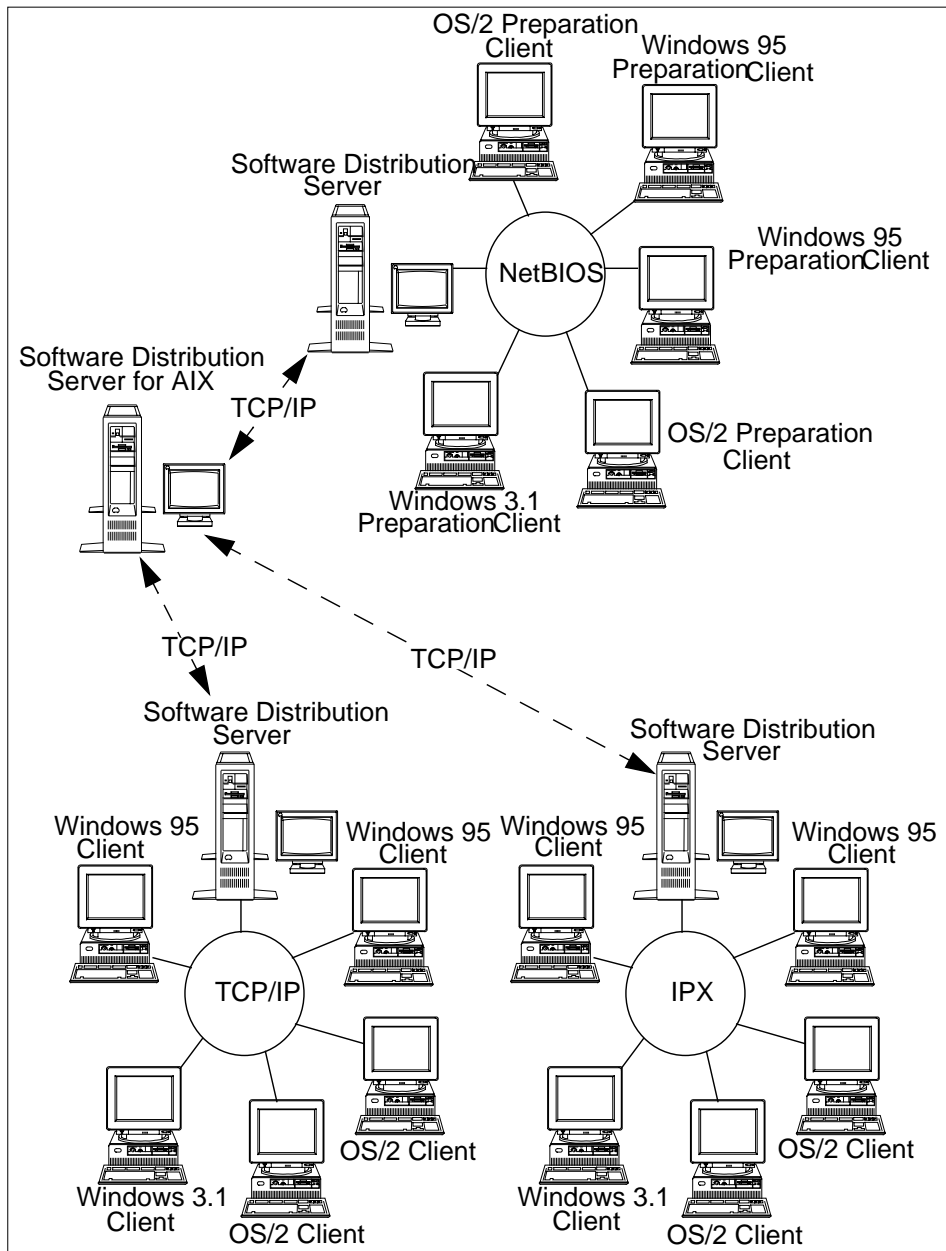


Figure 8. Software Distribution from AIX

Networks can also include remote connections to systems running any of the products in the Software Distribution/NetView Distribution Manager family.

2.5 Positioning LCU, NetView DM/2 and TME 10 SD 3.1.3 - Summary

This section provides information that can be used for evaluating either the LAN CID Utility or NetView DM/2 V2.1 as a software distribution manager. It focuses on what can be done by these products, not how it is done. Thus, this is not intended for a comparison of implementation details.

The table below summarizes the built-in features of LAN CID Utility, NetView DM/2 V2.1 and TME 10 SD V3.1.3.

Table 6. Positioning LCU, NVDM/2, and TME 10 SD 3.1.3 (Part 1 of 2)

Supported Features	LAN CID Utility	NetView DM/2 V2.1	TME 10 Software Distribution 3.1.3
Operating Systems at Target/Client	OS/2	OS/2 DOS/Windows	OS/2, Windows 95, Windows 3.1, Windows NT.
Systems environments	Stand-alone LAN	Stand-Alone LAN Host-connected LANs	Stand-Alone LAN Host-connected LANs
Place of Task Invocation	Local Workstation/Server	Local Server Remote Focal Point	Local Server Remote Focal Point
Remote Procedure Invocation	no	yes	yes
Scheduled Task execution	no	yes	yes
SDM supporting CID-enabled installation	yes	yes	yes
SDM supporting non-CID-enabled installation	no	yes	yes
Installation Modes	Lightly Attended	Unattended	Unattended
Pristine Client Installation	Lightly Attended	Lightly Attended	Lightly Attended
Central Tracking of Resources	no	yes	yes

Table 7. Positioning LCU, NVDM/2, and TME 10 SD 3.1.3 (Part 2 of 2)

Supported Features	LAN CID Utility	NetView DM/2 V2.1	TME 10 Software Distribution 3.1.3
ASCII/EBCDIC conversion	n/a	yes	yes
Automatic compression before transmission	n/a	yes	yes
Monitor status of data transmission	n/a	yes	yes
Autorecovery after line failure	n/a	yes	yes
Automatic disk space verification (product dependent)	no	no	no
Automatic backup before installation	no (possibility for full backup)	optional (for non-CID) no (CID enabled)	optional (for non-CID) no (CID enabled)

2.6 Overview of Software Distribution Managers

A set of clients controlled by a server is known as a *change control domain*. Clients in a server domain are called *local targets*, and clients in different domains are referred to as *remote targets*.

Simple Software Distribution systems include a server with its clients, a single domain. A more complex network can contain more than one interconnected server connected in a hierarchical structure and governed by a central manager system.

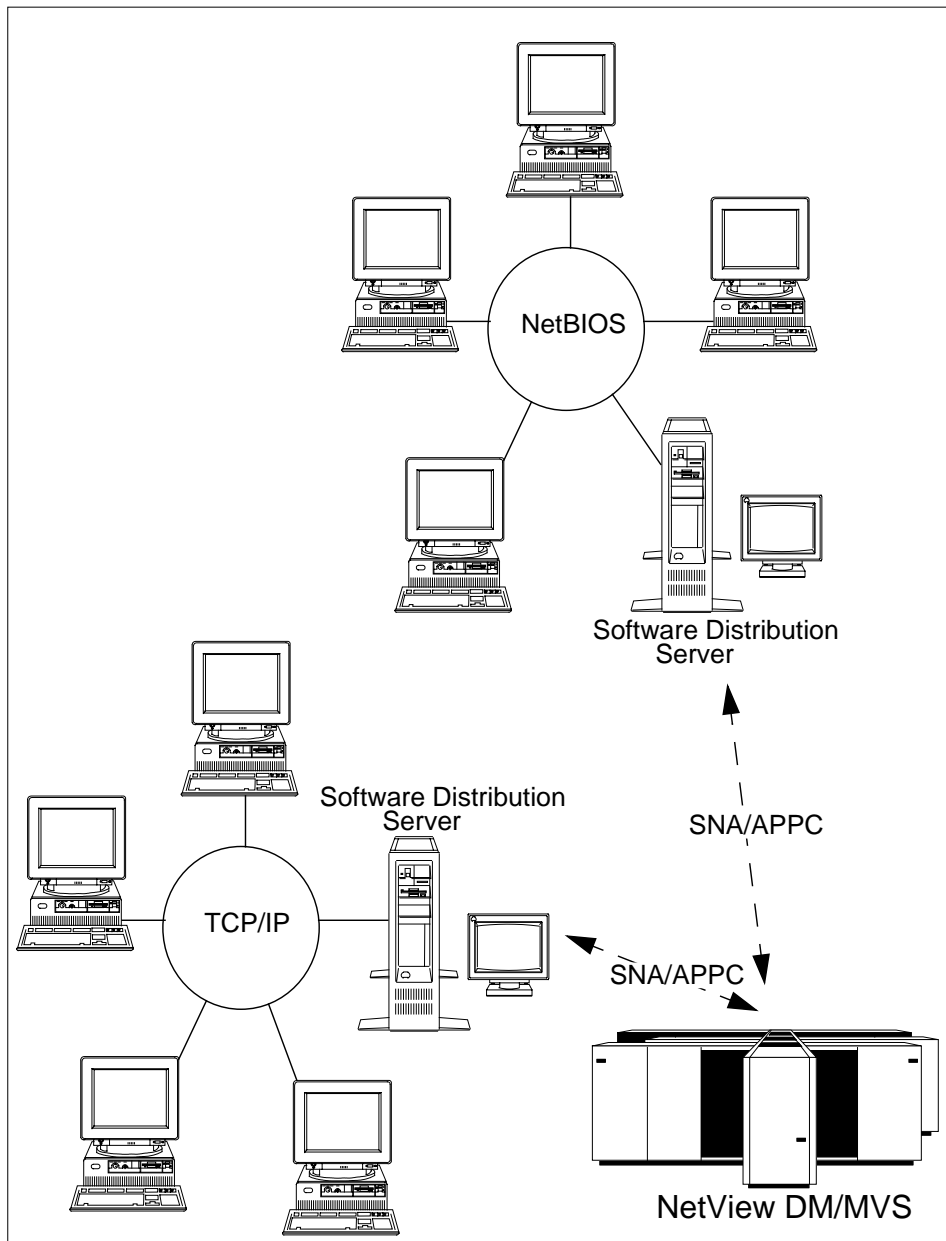


Figure 9. Complex Distribution Network through SNA/APPC Connection

Two different methods can be used to distribute software to the clients.

- Push Mode

The administrator at the server and the user at the client can initiate installation of software on a client workstation. With change control, the administrator is able to determine if changes are required and schedule the change control operations from the server.

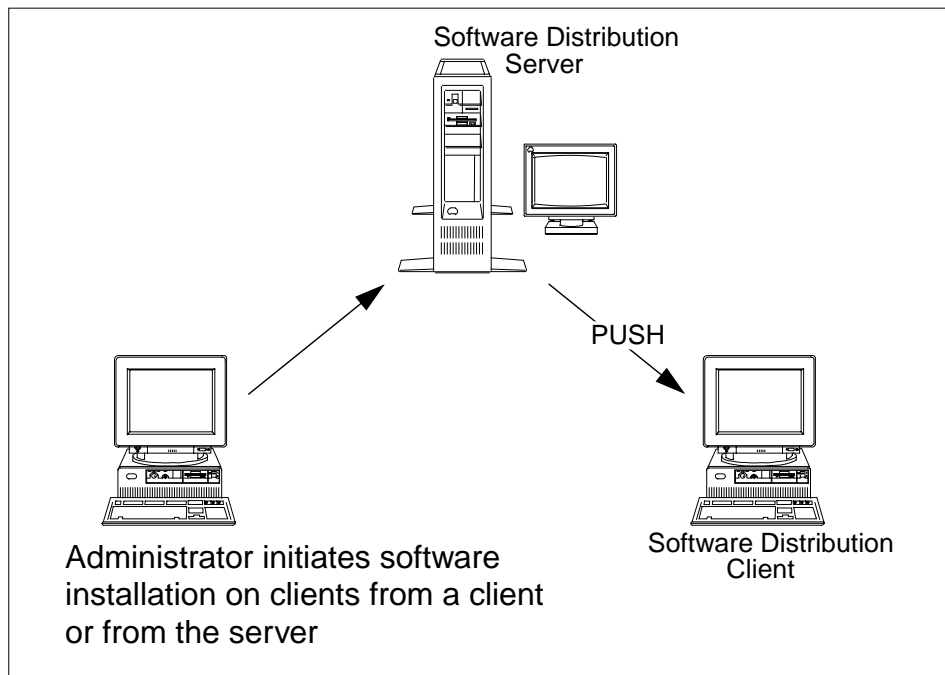


Figure 10. Software Distribution through Push Mode

- Pull Mode

A client workstation in pull mode is controlled by the user of the workstation. The user can install any software package that has been authorized to install on this workstation.

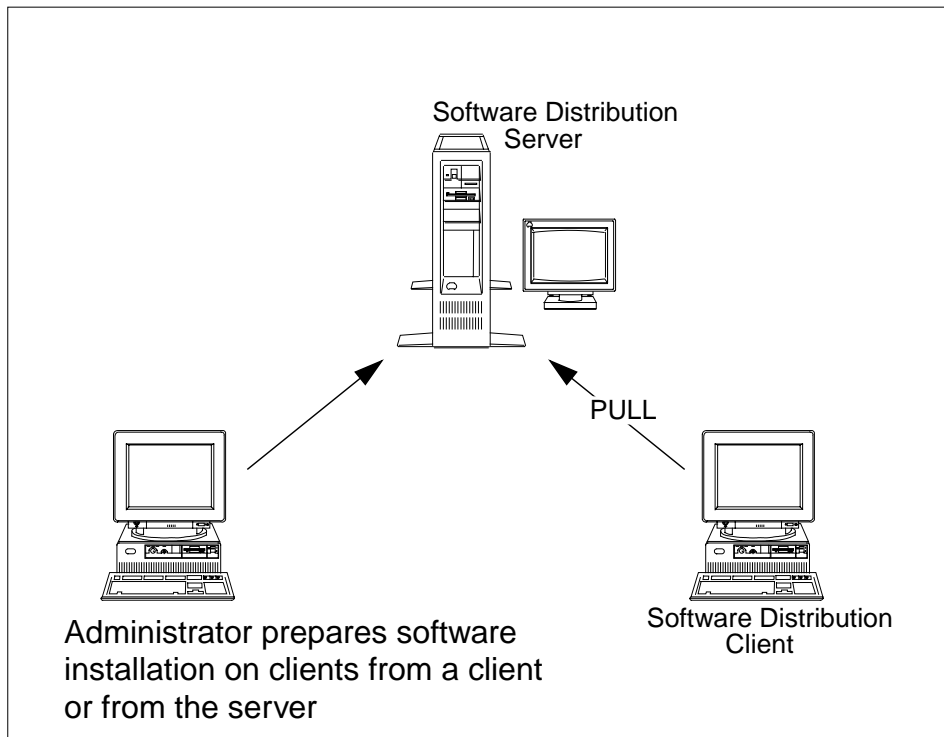


Figure 11. Software Distribution through Pull Mode

An active client is able to initiate operations on other clients in the network. A passive client can only have push operations performed on it.

Independent from the software distribution manager used, the following tasks can be performed on a client in either push or pull mode:

- Install a new software package on a client.
- Apply updates to currently installed software.
- Apply fixes to currently installed software.
- Back up to previous level of software.
- Accept changes permanently.
- Activate software.
- Uninstall one or more software packages installed previously.

Chapter 3. OS/2 Warp 4 - Installation Steps Local Install

This chapter illustrates the integrated install of OS/2 Warp 4. It is essential to understand the installation process of a local OS/2 Warp 4 install to effect an orderly remote install through software distribution managers. In comparison to previous OS/2 versions, OS/2 Warp 4 comes with a so-called Feature Installer (FI), that plays an important role in the installation process.

The actual installation process consists of several independent parts bound together and run in the background. The integrated installation hides the underlying structure and makes the actual installation process more transparent for the end-user.

The local installation is divided into three phases. The first phase is the text-based interface, which copies the needed files to start the base operating system in the graphical interface.

The second phase is the base operating system installation, including the selections for the networking components. The Feature Installer installs the new features of OS/2 Warp 4, such as Java, developer API extensions, and the registration tool. The MPTS product is installed during this phase as well.

The third phase includes the installation of the networking components, like TCP/IP, NetWare requester and LAN Distance client, according to the selections made during phase two. Because of the fact that the integrated install relies on the same procedures used in the CID installation, it is very important to understand the way how the different products are being installed.

3.1 Installation Diskettes

The three installation diskettes can be created from the CD-ROM if required. To create the diskettes, type

```
CDINST
```

from the CD-ROM's root directory. The command can be executed from OS/2 and DOS.

The diskettes are named Installation Diskette (Diskette 0), Diskette 1 and Diskette 2. The installation diskette contains OS/2 boot files, including OS2BOOT, OS2VER, OS2LDR, OS2KRNL1, and ABIOS files for micro channel-based machines. This diskette is always the same regardless whether it is used for local or remote installs and should not be modified.

Diskette 1 contains the CONFIG.SYS file to control the device drivers to be loaded. All the BASDEV drivers defined in the CONFIG.SYS reside on Diskette 1. These drivers include drivers for SCSI controllers, CD-ROM adapter cards, sound cards, and IDE devices, including CD-ROMs. Diskette 2 holds device drivers that Diskette 1 cannot hold due to lack of diskette space. Be aware, there is only one CONFIG.SYS file, and it always resides on Diskette 1.

OS/2 Warp 4 uses a new hardware detection process called *snooping*. Snooping is a method to detect the hardware installed and load the right device drivers. Hardware detection is not an easy task to implement because there is no common interface to detect ISA adapters. PCI and microchannel buses (MCA) are easier in that respect because the system BIOS is in charge to maintain a list of installed adapter cards. PCI bus implementation requires that the adapter card has to register itself in the BIOS when microchannel adapter identification is made with .ADF description file.

The way to detect adapters, in general, is to try to load the device driver. If the driver is loaded successfully, the adapter is installed. If the driver load fails, the adapter is not present. The problem of this procedure is that the system may hang due to a wrong device driver that was being loaded.

In a case when the booting process stops during processing Diskette 1, the reason may be as mentioned above. To find out if a device driver hangs the system, press **ALT-F1** when the machine is booted from the installation diskette and a small white bar appears in the upperleft corner with OS/2 printed beside it. The keystroke enables displaying the device driver to be loaded in the bottom of the screen. When the problematic device driver is known, the corresponding device driver line can be remarked from the CONFIG.SYS file. The SNOOP.LST file should be edited in the same manner. If the hardware is known, all the unnecessary device drivers can be removed or remarked from the CONFIG.SYS and SNOOP.LST to improve the system bootup time.

3.2 Local Install Phase 1

The first local install step is the text-based interface to select the installation type, installation drive and file system. The first phase copies the files needed to run the base operating system from the hard disk and creates a new CONFIG.SYS file for the next phase.

The OS/2 Diskette 1 contains the CONFIG.SYS file and device drivers. The CONFIG.SYS file includes statements to load the needed device drivers and

statements to invoke the installation process. The environment variables used to control the installation process are listed in Table 8 on page 41.

Table 8. The CONFIG.SYS Environment Variables in Phase 1(Part 1 of 2)

Variable	Initial Value	Used by	Explanation
CSFSTAGEDRIVE		FSERVICE	Directs the service tool to keep the locked files on the defined drive. Valid only when installing ServicePaks.
SOURCEPATH		RSPINST	Setup by SEINST.EXE if it is used. This variable will override the response file settings.
TARGETPATH		RSPINST	Setup by SEINST.EXE if it is used. This variable will override the response file settings.
COPYFROMFLOPPY		SYSINST2	Tells the system to copy known drivers from the A: diskette and to overwrite the files copied from the installation CD-ROM.

Table 9. The CONFIG.SYS Environment Variables in Phase 1(Part 2 of 2)

Variable	Initial Value	Used by	Explanation
SAVECONNECT	1	SYSINST2	Tells the system not to delete unknown files when install is done.
DISKETTESOURCE		SYSINST2	Tells the system to use installation diskette drive other than A:.
CDROMINST	1	SYSINST2	Defines, if the installation is run from CD-ROM or not.
OEMPROGRAM	\IBMINST\ NPCONFIG.EXE	SYSINST2	Points to the secondary installation program. During the first phase, NPCONFIG tries to detect the network adapter card installed.
EXITWHENDONE	1	SYSINST2	Tells SYSINST2.EXE not to display reboot message but return to parent program when completed (CONINST.EXE).
OS2_SHELL	CDBOOT.EXE		The program to start the installation process.
PROTSHELL	SYSINST1.EXE		Working environment to be loaded. SYSINST1 loads the installation environment.

By default, all these variables are not set in the CONFIG.SYS file. The most commonly used additional parameters are COPYFROMFLOPPY and SAVECONNECT.

CDBOOT.EXE is the program that starts the installation process and then calls CONINST.EXE that calls SYSINST2.EXE, as shown in Figure 12 on

page 43. CONINST.EXE tries to detect the CD-ROM used and the installed network adapter card. SYSINST2.EXE queries the installation type, runs FDISK and formats the installation partition as required. SYSINST2 copies all the needed files to start the next step of installation from the hard disk in the graphical interface. The copied files are device drivers and OS/2 system files, like OS2KRNL and OS2LDR. The first step also sets the needed environment variables for installation phase 2. SYSINST2.EXE returns the control for CONINST.EXE after copying files. CONINST tries to detect the installed network adapter card and saves the information in a file SNOOP.OUT in the installation drive root directory. In the end, CONINST prompts the user to remove any diskettes and press **Enter** to restart the system.

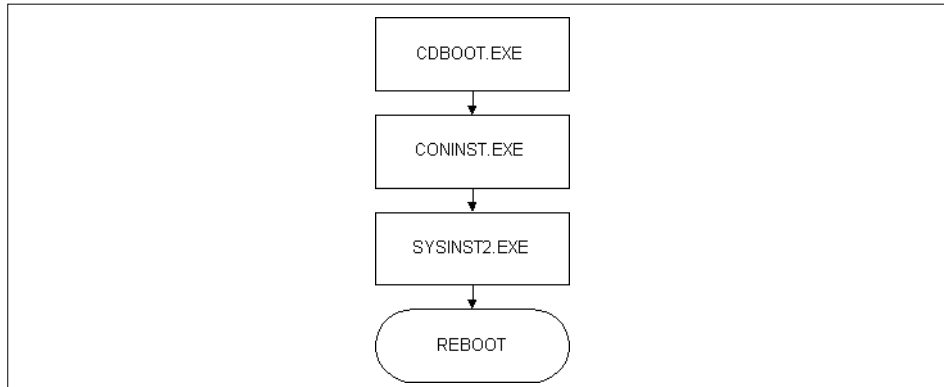


Figure 12. Local Install Phase 1

If any extra device drivers for hardware are needed, the device drivers have to be copied to the OS/2 Installation Diskette 1, and the corresponding statements to load the drives have to be added in the CONFIG.SYS. The COPYFROMFLOPPY and SAVECONNECT variables should be set equal to 1 to make sure that the files are being copied to the hard disk. COPYFROMFLOPPY=1 tells the system to copy all known drivers from the diskette and overwrite the files copied from the CD-ROM. SAVECONNECT=1 tells install program to keep the connection files after install is finished. Normally, the install will copy all unknown files from the diskette to the hard drive and will migrate CONFIG.SYS statements that are unknown. Then after install completes, it deletes these unknown files. If SAVECONNECT is set, the system does not delete these unknown files when install is done.

3.3 Local Install Phase 2

Installation phase 2 installs the rest of the operating system, including IBM Works, VoiceType, Java support, and other additional software. This really is something new that is only included in OS/2 Warp 4: The Feature Installer (FI). Executed by its command line interface, CLIFI, the feature installer uses the \OS2\INSTALL\FIBASE.RSP along with a partial response file to install components such as Developer API Extensions (DAX), Java 1.02 or even the registration tool (ART). Later on, the Feature Installer can be used to uninstall components installed by FI. For example, at a local workstation, open the drives folder and then expand the \OS2\INSTALL tree. You will see a directory called **Installed Features**. Double-click on it and check for options. You can use this method to uninstall the registration tool properly.

When the operating system boots, there is no desktop; PMSHELL starts and creates the maintenance desktop for installation as shown in Figure 13. \OS2\INSTALL\INSTALL.INI contains the information needed to create the desktop. Maintenance desktop is very simple: Only the most important programs are included, like OS/2 command prompts and some system tools, like editors. Maintenance desktop includes the startup folder, and the ISHIELD.EXE program is run from there. The main installation program, INSTALL.EXE, is invoked from by PMSHELL when the desktop is up and running if the `RUNINSTALL` variable is set. The `RUNINSTALL` variable and all the other environment variables are declared later on. ISHIELD.EXE is running throughout the whole installation, and its purpose is to provide a common interface to all the separate software component installation programs running in the background.

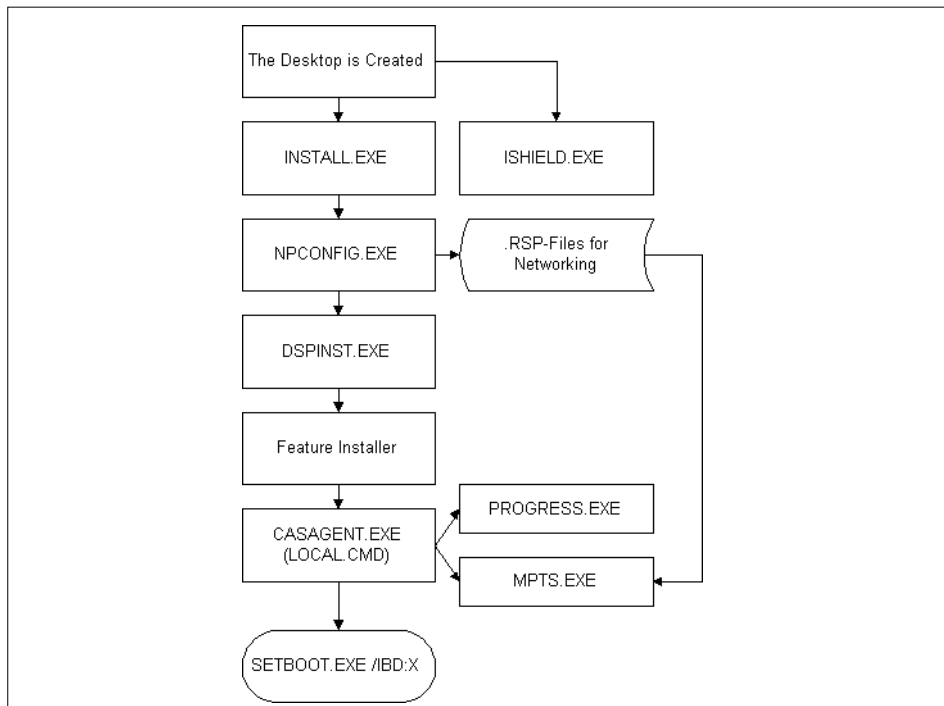


Figure 13. Local Install Phase 2

The local install phase 2 starts the basic OS/2 operating system installed to the hard disk during phase 1. CONFIG.SYS file contains several environment variables to transfer information between phases 1 and 2 and to invoke the required installation programs. Here is a sample CONFIG.SYS file:

```

IFS=\OS2\INSTALL\HPFS.IFS /AUTOCHECK:D
DEVINFO=SCR,VGA,\OS2\BOOT\VIOTBL.DCP
SET VIDEO_DEVICES=VIO_VGA
SET VIO_VGA=DEVICE(BVHGA)
PROTSHELL=\OS2\PMSHELL.EXE
SET USER_INI=\OS2\INSTALL\INSTALL.INI
SET SYSTEM_INI=\OS2\INSTALL\INSTALL.INI
SET OS2_SHELL=\OS2\CMD.EXE
SET AUTOSTART=PROGRAMS,TASKLIST,FOLDERS
SET RUNWORKPLACE=\OS2\PMSHELL.EXE
SET COMSPEC=\OS2\CMD.EXE
LIBPATH=.\OS2\INSTALL;\OS2\DLL;\OS2\MDOS;\C:\IBMINST;
SET PATH=.\OS2;\OS2\BOOT;\OS2\SYSTEM;\OS2\INSTALL;\OS2\MDOS;\C:\IBMINST;
SET DPATH=.\OS2;\OS2\BOOT;\OS2\SYSTEM;\OS2\INSTALL;\C:\IBMINST;
SET PROMPT=[ $P ]
SET HELP=\OS2\HELP\C:\IBMINST;
PAUSEONERROR=NO
BUFFERS=90
IOPL=YES
DISKCACHE=D,LW
MAXWAIT=3
  
```

```

MEMMAN=SWAP,PROTECT
THREADS=128
DEVICE=\OS2\BOOT\DOS.SYS
DEVICE=\OS2\BOOT\PMDD.SYS
DEVICE=\OS2\BOOT\OS2CDROM.DMD /Q
SET KEYS=ON
PROTECTONLY=YES
BASEDEV=IBMKBD.SYS
SET DESKTOP=<WP_MAINT>
DEVICE=\OS2\BOOT\POINTDD.SYS
DEVICE=\OS2\BOOT\MOUSE.SYS
CODEPAGE=850
COUNTRY=001,\OS2\SYSTEM\COUNTRY.SYS
DEVINFO=KBD,US,\OS2\KEYBOARD.DCP
SWAPPATH=C:\OS2\SYSTEM 64 2048
BASEDEV=PRINT01.SYS
BASEDEV=IBM1FLPY.ADD
BASEDEV=IBM2FLPY.ADD
BASEDEV=IBM1S06.ADD
BASEDEV=IBMINT13.I13
BASEDEV=OS2DASD.DMD
BASEDEV=DAC960.ADD
DEVICE=\OS2\BOOT\TESTCFG.SYS
SET INSTALLTYPE=ADVANCED
SET SOURCEPATH=E:\OS2IMAGE
SET DISKTYPE=2
SET FDISKETTESOURCE=0
SET FIRSTDISK=12
SET RAMABIOS=0
SET NUMDISKS=28
SET TARGETPATH=C:
SET SAVECONNECT=1
SET CDROMINST=1
SET OEMPROGRAM=ibminst\npconfig.exe
SET MACHINE=1
SET VIODISPLAY=4
SET VIOADAPTER=8
SET VIOMEM=262144
SET MODEL=252
SET SUBMODEL=1
SET PLANARID=8
SET MOUSE=0
SET MOUSECOM=0
SET FULLINST=CAB$3$4$
SET DUALBOOT=0
SET HPFSREQ=1
SET FORMAT=1
SET RUNINSTALL=C:\OS2\INSTALL\INSTALL.EXE
SET OLDUSER_INI=C:\OS2\OS2.INI
SET OLDSYSTEM_INI=C:\OS2\OS2SYS.INI
REM -----
REM   DOS support for SVGA
REM -----
FILES=20
BREAK=OFF
PROTECTONLY=NO
SHELL=\OS2\MDOS\COMMAND.COM \OS2\MDOS /P
FCBS=16,8
RMSIZE=640
DOS=LOW,NOUMB
DEVICE=\OS2\MDOS\VSVGASYS.SYS
BASEDEV=OS2SCSI.DMD
BASEDEV=XDFLOPPY.FLT

```

```
IFS=CDFS,IFS /Q
BASEDEV=CHINCDS1.FLT
BASEDEV=HITCDS1.FLT
BASEDEV=NECCDS1.FLT
BASEDEV=SONYCD1.FLT
BASEDEV=TOSHCDS1.FLT
BASEDEV=IBMIDEC.D.FLT
BASEDEV=SONY535.ADD
BASEDEV=LMS206.ADD
REM 154569 BASEDEV=LMS205.ADD
BASEDEV=MITFX001.ADD
BASEDEV=SBDC2.ADD
BASEDEV=SONY31A.ADD
SET OEMPROGRAM=\IBMINST\NPCONFIG.EXE
SET EXITWHENDONE=1
DEVICE=\OS2\BOOT\REFPART.SYS
```

In this phase, the hardware configuration is not yet specified; so all the CD-ROM device drivers are included to access the installation media. The environment settings used for this phase of installation are listed in Table 10 on page 48.

Table 10. The CONFIG.SYS Environment Variables in Phase 2 (Part 1 of 3)

Variable	Initial Value	Legal Values	Explanation
USER_INI	\OS2\INSTALL\ INSTALL.INI		The initial USER_INI file.
SYSTEM_INI	\OS2\INSTALL\ INSTALL.INI		The initial SYSTEM_INI file.
DESKTOP	<WP_MAINT>		The desktop to be used is the maintenance desktop.
INSTALLTYPE	ADVANCED	EASY, ADVANCED	The installation type selected during the first installation phase.
SOURCEPATH	D:\OS2IMAGE		Source path for the installation files, where D: is the CD-ROM drive letter.
DISKTYPE	2		
FDISKETTESOURCE	0	0, 1	0 equals to floppy A: and 1 to floppy B:.
FIRSTDISK	12		The diskette number to start installation from.
RAMABIOS	0		With a non-Microchannel machine, the value is 0.
NUMDISKS	28		The total number of diskettes for installation. The diskettes handled are from FIRSTDISK to NUMDISK
TARGETPATH	C:	Any local drive letter	The installation drive letter.

Table 11. The CONFIG.SYS Environment Variables in Phase 2 (Part 2 of 3)

Variable	Initial Value	Legal Values	Explanation
MACHINE	1	1, 2	Machine bus type. 1 = ISA 2 = Microchannel
VIODISPLAY			The detected display type.
VIOADAPTER			The detected video adapter type.
VIOMEM			The amount of video memory installed.
MODEL			The machine model. The information is fetched from the machine BIOS.
SUBMODEL			The machine type. The information is fetched from the BIOS.
PLANARID			
MOUSECOM	0	0, 1	Boolean type value to identify if the mouse is a PS/2- or a serial mouse.
FULLINST	CAB\$3\$4\$		Tells that this installation is a new install. Do not change this parameter.
DUALBOOT		0, 1	Equals to 1 if the Dual-Boot system was installed.
HPFSREQ	1	0, 1	If HPFS support is required or not.

Table 12. The CONFIG.SYS Environment Variables in Phase 2 (Part 3 of 3)

Variable	Initial Value	Legal Values	Explanation
FORMAT		0, 1	Equals to 1 if the installation partition was formatted.
RUNINSTALL	C:\OS2\INSTALL\INSTALL.EXE		The installation program that is activated after the desktop is up and running.

All these values are set by the first installation step. Under normal conditions there is no need to change these values. If these values are changed anyway, the installation process may fail.

INSTALL.EXE is the main program to start the installation as shown in Figure 5 on page 46. INSTALL.EXE first runs hardware detection to find the installed adapters, including SCSI, video, sound, and multimedia adapters. The hardware detection is actually done by small help programs invoked from INSTALL.EXE, but the result codes are returned to and handled by INSTALL.EXE.

After the hardware detection has completed, INSTALL.EXE gives user the selection screens to verify the hardware profile and change it if needed. The configuration selection consists of three screens, of which two are for hardware and one for software components. The first of the three system configuration screens is shown in Figure 6 on page 45. INSTALL.EXE creates a response file, \IBMINST\RSP\LOCAL\USER.RSP, according to the selections made during this phase. The file can be used later on for new local or CID installations.

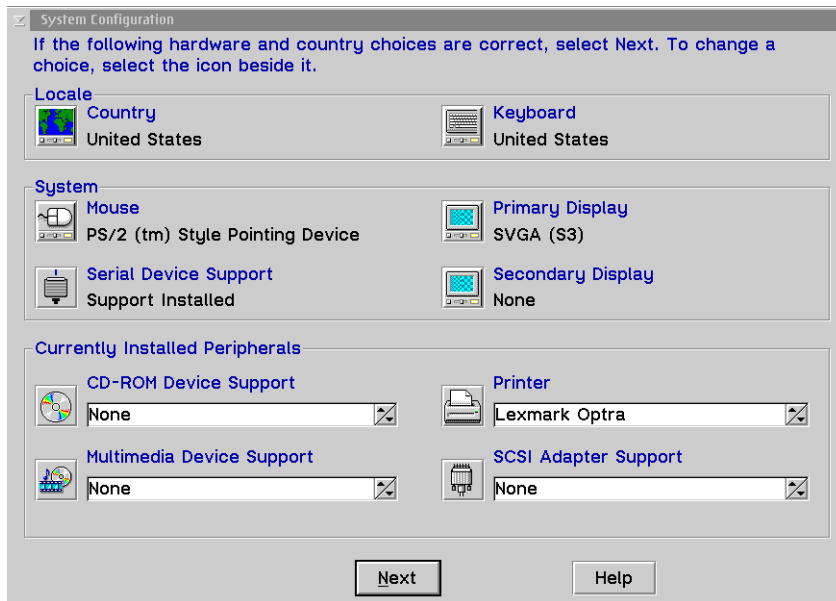


Figure 14. The System Configuration Screen

After the software component selection, the NPCONFIG.EXE is started, and if the advanced installation was selected, the main screen is displayed as shown in Figure 15 on page 52. NPCONFIG.EXE queries the user for the network configuration, including network adapter settings, protocol settings and other parameters, like TCP/IP configuration information, if selected to be installed. NPCONFIG.EXE then creates the response files for networking and system management products included in the Merlin package. The response files are stored in the directory C:\IBMINST\RSP\LOCAL, where C: is the installation drive. NPCONFIG.EXE does not install anything; it just creates the response files and calls the needed programs to continue the installation process.

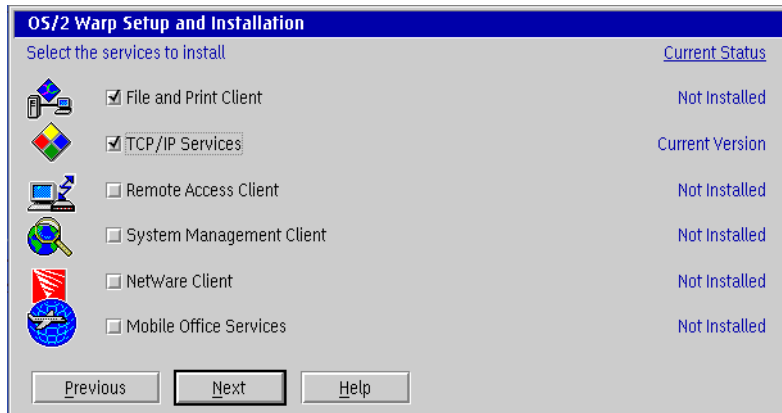


Figure 15. The Network Component Installation Screen

After NPCONFIG.EXE is run and the response files have been generated, the display driver install program (DSPINSTL.EXE) is run. Display driver install installs the right video driver or the VGA driver if the video controller chipset is not supported by Merlin by default. DSPINSTL leaves the screen resolution to VGA (640x480) even if a super-VGA chipset was found.

The next step in the flow is the Feature Installer, as shown in Figure 13 on page 45. The Feature Installer installs the OS/2 Warp 4 BonusPak applications including VoiceType, BonusPak, Java support, and all the other non-networking components included in the OS/2 Warp 4 package. Feature Installer is embedded in the INSTALL.EXE and INSTALL.DLL library. The Feature Installer is something new to OS/2 Warp version 4 compared to the earlier version of OS/2. Feature Installer uses a new response file format and can be used to install generally any kind of software.

The last product to be installed during this step is MPTS. MPTS is installed even if there is no network adapter present; in this case, the loop-back driver is installed. MPTS installation uses the response file created by NPCONFIG.EXE earlier during the installation.

All the networking components including MPTS are installed by CASAGENT.EXE running LOCAL.CMD REXX script. LOCAL.CMD is described in detail later on in “The LOCAL.CMD Command File (IBMINST\RSP\LOCAL)” on page 58. PROGRESS.EXE displays the progress bar on the screen to give the user more information about the status of installation.

Because no diskettes are used during this phase, the machine can be automatically booted after this phase of installation is completed. The reboot is done by calling SETBOOT.EXE with the parameter /IBD:C, where C is the installation drive letter. If the Boot Manager is installed, it will be bypassed, and the installation continues from phase 3.

3.4 Local Install Phase 3

Local install phase 3 installs all the components related to networking and system management. This phase starts with locked files processing. The locked files are processed by a device driver, \OS2\INSTALL\IBMLANLK.SYS, and a run statement, \OS2\INSTALL\IBMLANLK.EXE, in the CONFIG.SYS:

```
DEVICE=C:\OS2\INSTALL\IBMLANLK.SYS C:\OS2\INSTALL\IBMLANLK.LST
RUN=C:\OS2\INSTALL\IBMLANLK.EXE C:\OS2\INSTALL\IBMLANLK.LST
```

New objects are created on the desktop when the desktop is started.

The main installation is started from the STARTUP.CMD file. Here is a sample STARTUP.CMD file:

```
@ECHO OFF
C:\OS2\INSTALL\ISHIELD.EXE
C:\OS2\INSTALL\WRAPPER.EXE
CALL C:\CHKFORCD.CMD
C:\CID\LOCINSTU\CASAGENT.EXE /REQ:LOCAL /CMD:C:\IBMINST\RSP\LOCAL /D
                        /L1:C:\IBMINST\LOGS\LOCINSTU\LOCAL.L2

CMD
EXIT
```

As in phase 2, ISHIELD.EXE is started first. WRAPPER.EXE takes keyboard and mouse control so that the installation cannot be cancelled by user action during processing. LOCAL.CMD checks to see if the WRAPPER.EXE and ISHIELD.EXE are running; if not, LOCAL.CMD launches these programs.

The CASAGENT.EXE statement is started with several parameters.

/REQ:LOCAL defines the LAN CID Utility client name of the workstation. This parameter is discussed in deeper detail later in the LCU section.

/CMD:C:\IBMINST\RSP\LOCAL\LOCAL defines the LCU command file to be used.

The file must be a REXX command file. The /D parameter causes CASAGENT.EXE to use the DEFAULT.CMD command file if the one defined by the /CMD option is not found. The same directory defined by the /CMD option will be used with the /D parameter. /L1:C:\IBMINST\LOGS\LOCINSTU\LOCAL.L2 parameter creates an installation log file.

The installation goes on with network product installations starting from FFST (First Failure Support Technology; a software architecture), as shown in Figure 16 on page 54. All the product installations are invoked from the LOCAL.CMD command file. LOCAL.CMD calls every product's own installation program with the needed parameters, but this structure is hidden under the ISHIELD.EXE program. The boxes with continuous borders are considered to be part of the basic network installation, and boxes with dashed border lines are for extra functionality. Any combination, or all the components, can be installed.

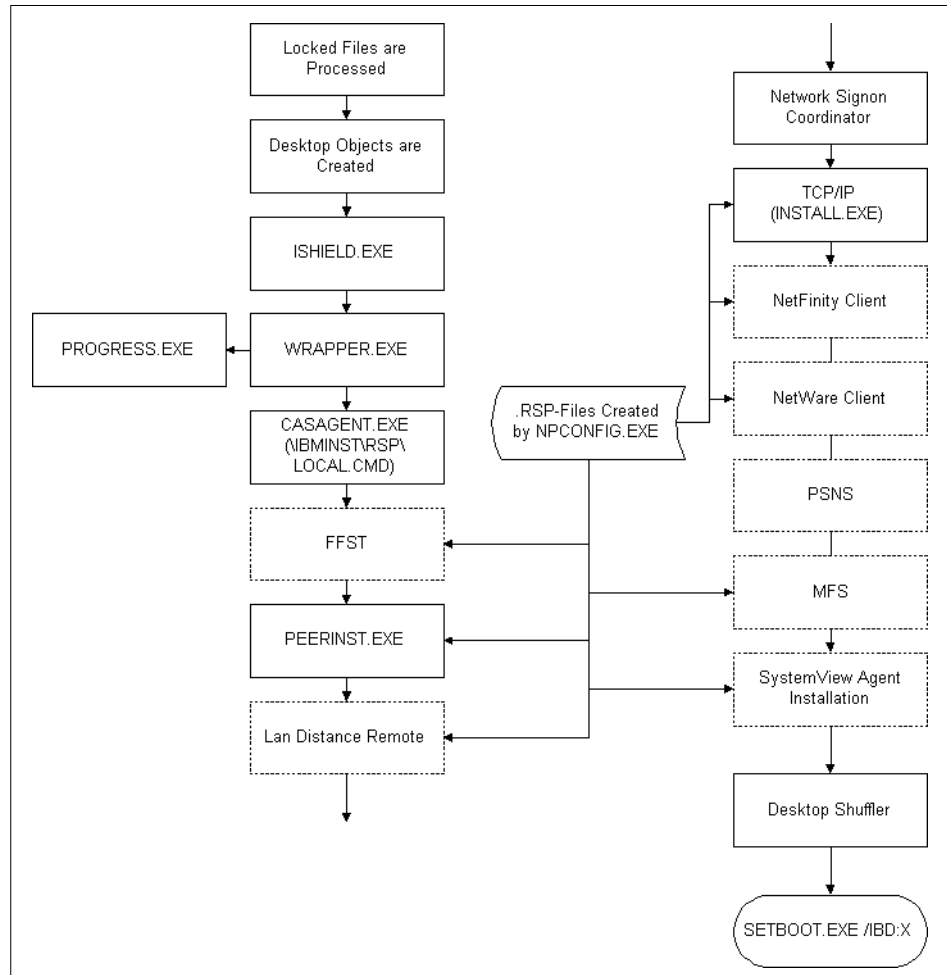


Figure 16. Local Install Phase 3

After the selected components have been installed, the Desktop Shuffler is run to arrange the desktop, move the created icons to the right folders, and rename the icons when needed. The machine is rebooted automatically using the `SETBOOT.EXE` command. After this boot, the locked files are handled, and the installation is ready.

The actual installation programs for each component during phase 3 are listed in Table 13, where D: equals to the CD-ROM drive letter, and C: is the OS/2 installation drive letter.

Table 13. Installation Programs (Part 1 of 2)

Product	Installation Program	Product Description
FFST	C:\IBMINST\RSP\LOCAL\INSTFFST.CMD	Utilities for problem determination and solving.
LAN Requester / Peer	D:\CID\IBMPEER\PEERRMT	The basic product to get LAN support for file and print servers and for peer networks.
LAN Distance Remote	D:\CID\LDREM\INSTALL	Drivers to support network functions over a telephone line.
Network SignOn Coordinator/2	D:\CID\NSC\INSTALL.EXE	Utility to keep passwords in sync.
TCP/IP	D:\CID\TCPAPPS\INSTALL	TCP/IP applications, including Utmilite, Web Explorer and DOS/Windows support for TCP/IP.

Table 14. Installation Programs (Part 2 of 2)

Product	Installation Program	Product Description
NetFinity	D:\CID\NETFIN\NETFINST.EXE	NetFinity client to support remote management.
NetWare Client	C:\IBMINST\NPNWINST	NetWare client for OS/2. Enables connections to Novell NetWare servers and contexts.
Personally Safe & Sound	D:\CID\PSNS\PSNSINST	Backup program.
Mobile File Sync	D:\CID\MFS\INSTALL.EXE	Utility to keep remote and local files in sync.
SystemView Agent	D:\CID\SVAGENT\INSTALL.COMD	SystemView Agent program for remote management.

All the installation programs are run from the CD-ROM, except NetWare client installation and FFST installation that are run from the local drive's \IBMINST directory. None of the components listed here is required for installing or running the basic operating system, but you need at least LAN or Novell NetWare requester to get access to network resources like printers or directory shares from the file server or servers.

3.5 The Installation Directory Structure

The directory structure for the OS/2 Warp 4 installation is quite simple. A sample structure when starting installation phase 2 is shown in Figure 17 on page 57. The entire directory structure is not included, only those branches relevant to the installation procedure.



Figure 17. OS/2 Warp 4 Directory Structure

The root directory is used to store some temporary files, but all these files are deleted before the installation finishes. Also, all the device drivers are loaded from the root directory instead of the \OS2\BOOT directory during phases two and three.

The \CID\LOCINSTU directory contains all the LCU-related file, including CASAGENT.EXE. \IBMINST is the most important directory for the installation process. The directory contains executable files like NPCONFIG.EXE and ISHIELD.EXE, and smaller help programs to enable hardware detection, and executables to complete some other tasks related to the installation procedure. All the response files and LOCAL.CMD REXX script files are stored in the \IBMINST\RSP\LOCAL subdirectory.

The installation log files are created under \IBMINST\LOGS and under a corresponding directory. INSTALL.EXE itself is run from the \OS2\INSTALL directory.

3.6 The Local Install Key Files

The following section introduces the local installation key files including LOCAL.CMD, NPCONFIG.EXE and response files created by NPCONFIG.EXE.

3.6.1 The INSTALL.EXE Executable (\OS2\INSTALL)

INSTALL.EXE is the main executable program for installation. INSTALL.EXE installs the base operating system and includes the Feature Install code to install IBM Works, VoiceType, Java support, and other applications included in the OS/2 Warp 4 package.

INSTALL.EXE uses functions from INSTALL.DLL dynamic link library. INSTALL.EXE can be run anytime to install new or reinstall existing components or change the installed hardware configuration. INSTALL.EXE does not take any parameters when run from the command line. INSTALL.EXE can be run from desktop by clicking [**OS/2 System — System Setup — Install/Remove — Selective Install**].

3.6.2 The CASAGENT.EXE Executable (\CID\LOCINSTU)

The CASAGENT.EXE is part of the LAN CID Utility that invokes the installation procedure written in the LOCAL.CDM file.

3.6.3 The LOCAL.CMD Command File (\IBMINST\RSP\LOCAL)

The LOCAL.CMD file is the key file to OS/2 Warp 4 local installation. It includes unattended, response-file-based installation procedures to install networking and system management products. The basic idea with LOCAL.CMD is that it calls products' installation procedures and passes response file name, log file name and other required parameters.

LOCAL.CMD also passes installation information for PROGRESS.EXE, which displays the progress indicator and the product name being installed. The same principles used in LOCAL.CMD apply to LCU installation over the network. The main difference is that the source files are read from a redirected network drive instead of the CD-ROM.

3.6.4 The NPCONFIG.EXE Executable (IBMINST)

NPCONFIG.EXE provides the graphical interface used to select the networking components to be installed, and allows the user to change the networking configuration. NPCONFIG.EXE creates the RSP files for system management and network products and edits the LOCAL.CMD file. The installation process is handled by LOCAL.CMD. NPCONFIG.EXE is only used to configure the products and invoke the installation process.

NPCONFIG.EXE can be run with option /RSP to create the response files only without installing anything. The response files are stored in the directory \IBMINST\RSP\REMOTE.

3.6.5 The Response Files (\IBMINST\RSP\LOCAL)

NPCONFIG.EXE creates networking product response files during the installation, and these files are used later on to install the products without user input from LOCAL.CMD. The directory for these response files is \IBMINST\RSP\LOCAL. All the response files used under installation are listed in Table 15.

Table 15. Response Files Used in the Local Installation

Response File	Explanation
FIBASE.RSP	FIBASE.RSP is the response file Feature Installer uses. FIBASE.RSP includes all the additional components' definitions, including these for VoiceType, Java support and IBM Works.
MPTS.RSP	MPTS is installed in any case, even though there is no network adapter installed or no network components are installed. The response file includes information about protocols, adapters to be installed and bindings for protocols. MPTS.RSP outlook may vary quite a lot depending on the protocols and network adapter selected.
OS2PEER.RSP	OS2PEER.RSP file is created if the File and Print Client was selected to be installed as shown in Figure 6 on page 48. OS2PERR.RSP controls the LAN requester and peer settings, like workstation NetBIOS name and domain/workgroup name, as well as services to be installed.
TCPAPPS.RSP	TCPAPPS.RSP file is created if the TCP/IP Services was selected to be installed. The file contains TCP/IP configuration information and the list of the TCP/IP applications to be installed.

Response File	Explanation
USER.RSP	The USER.RSP response file is created by the OS/2 installation program INSTALL.EXE, and it includes the selections made for local install of the base operating system. The file can be used for future local or remote installs to install the same features on other machines without user input. The response file format and keywords are described in detail in "OS/2 Warp 4 - Base Operating System" on page 182.
LDR.RSP	LAN Distance Remote response file. The file includes the destination drive for installation, remote number to be dialed, modem port, and modem type definition file reference.
MFS.RSP	The Mobile File Sync program is part of the Mobile Office services. The file contains the product installation drive and directory and some other installation information.
NETFIN.RSP	The response file includes the NetFinity parameters, like enabled protocols, system name and optional parameters.
NPNWINST.RSP	The file includes the Novell NetWare preferred server name and if using version NetWare 4.0, also the default context name.
NSC.RSP	Network SignON Coordinator/2 response file includes the NSC installation directory and some other installation information.
NWMPTS.RSP	When using NetWare client, the ODI support if needed. NWMPTS.RSP includes the needed information to install the ODI2NDI protocol manager driver into MPTS.
SVAGENT.RSP	SystemView Agent response file includes installation directory and information about protocols to be enabled.

3.7 FixPak Installation

OS/2 Warp 4 FixPaks not only update the operating system, they also come with new features to OS/2 Warp 4, such as Java performance improvements, Graphics Adapter Device Drivers (GRADD), a new OS/2 Warp 4 Registry, and so forth. At the time this redbook was written, FixPak 4 and FixPak 5 for OS/2 Warp 4 (US English) were available on the Internet and intranet. Regardless of the number scheme, FixPaks that are run against the base operating system use the same remote installation techniques. Therefore, this section applies to all future OS/2 FixPaks.

Internet users can obtain OS/2 FixPaks from either Web site:

<http://service5.boulder.ibm.com/pspsfixpk.nsf/>
<ftp://ftp.software.ibm.com/ps/products/os2/fixes/v4warp/>

IBMers can take advantage of IBM's high speed intranet. Point your Web browser to the following Web site:

<http://os2service.austin.ibm.com>

OS/2 FixPaks do not have to be installed through diskettes. They can be applied directly from the Internet. To make use of the Netscape's and Web Explorer's RSU (Remote Software Update) feature, point your browser to this Web site:

- Internet:

<http://ps.boulder.ibm.com/pbin-usa-ps/getobj.pl?/pdocs-usa/softupd.html>

- Intranet:

<http://os2service.austin.ibm.com/intsoftupd.html>

You will have to select the right language directory and then the proper FixPak-level directory.

Note: When you update your OS/2 Warp 4 system through the Internet, you will be prompted to either delete the FixPak CID tree or leave it as it is. It is wise to save and copy the tree to the code server for later distribution to other workstations.

In summary, there are three ways to install a FixPak on the system:

1. Installation from diskettes
2. Installation directly from the Internet using RSU
3. Installation through the LAN from a software distribution server

The diskette installation can be done either from the Workplace shell using the graphical interface or by booting the system using Corrective Service Facility (CSF) disks (often called kicker diskettes) and installing the FixPak using text-based interface. The main difference between these two methods is that when the system is booted using the CSF diskettes, all the device drivers are loaded from the diskette, and there are no open files from the hard drive. When the Workplace Shell (WPS) is running, there are locked files that cannot be serviced at runtime. Service of the files in use has to be deferred and handled by the locked file device driver during the next boot.

The kicker diskettes can be downloaded from the following Web site:

```
ftp://ftp.software.ibm.com/ps/products/os2/fixes/wkickr/
```

The kicker diskette images are packed into a zip file, that has to be downloaded and unzipped.

The purpose of CSF is to apply a ServicePak or FixPak for the OS/2 operating system. Each product related to OS/2, for example the base operating system, Peer Services, TCP/IP or MPTS, that has to be serviced by a maintenance update, has a "syslevel" file. This syslevel file is installed with each product. For example, the OS/2 base operating system has a syslevel file named SYSLEVEL.OS2 that is installed in the \OS2\INSTALL directory. When maintenance is installed for a product, the corresponding syslevel files are updated to reflect the new "syslevel". The CSF uses these syslevel files to identify the products on the system and to verify that the products will not be "downleveled" by installing the maintenance.

When the CSF installs maintenance for a product, it must determine what directories are associated with the product. For each of the products serviced there is a set of default directories. These are the directories that would normally be serviced for this product. A ServicePak for the OS/2 base operating system services the root directory, the OS/2 directory, and all subdirectories of the OS/2 directory.

Getting Started With Kicker

The most important Corrective Service Facility files can be found on the second Kicker diskette. These Kicker diskettes are a pair of bootable diskettes that are used to service OS/2 systems.

FSERVICE.EXE (used for remote unattended installation)

SERVICE.EXE (used for attended installation from the GUI)

RESPONSE.FIL (sample response file covering multiple update scenarios)

Be sure that you are using the appropriate release of FSERVICE to apply your Service Pak.

3.7.1 The SERVICE Command

`SERVICE` is a command to install a ServicePak or FixPak using a graphical tool. Insert the first CSD diskette into drive A: and run the `SERVICE` command to start the system refresh. We recommend closing all applications running on the system being serviced.

Note

The `SERVICE` command will not work from redirected drives. `SERVICE` will only work from "removable" media such as floppy diskette and CD-ROM. `FSERVICE` is designed to work from floppy diskette, CD-ROM, hard drive, LAN drive, and so on.

Corrective Service Facility (CSF) displays the product and version information and queries for the media path to be used. After reading the first ServicePak diskette, you are prompted to select the products to service from the serviceable products list, as shown in Figure 18.

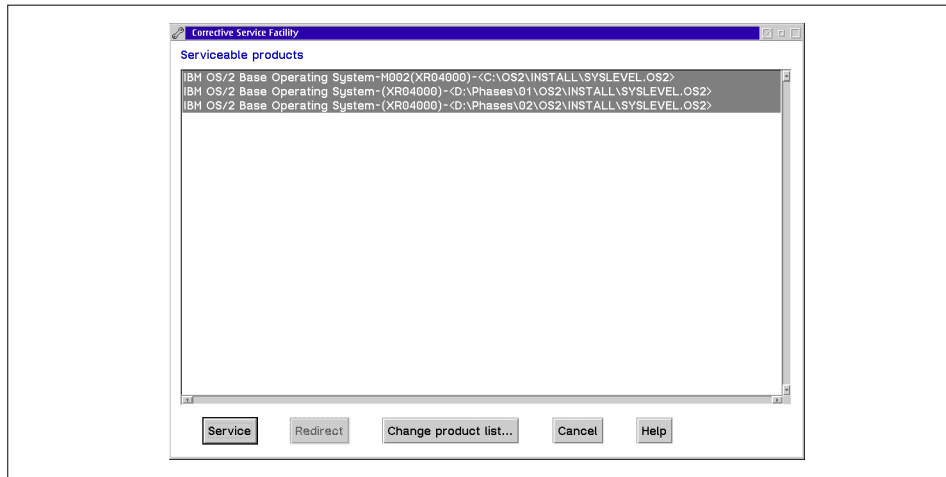


Figure 18. Serviceable Products on The System

After selecting the products to service, you are prompted for the archive and backup directories. If this is the first OS/2 Warp FixPak you have applied to this system, then enter the path to the ARCHIVE directory where a copy of replaced files will be stored (for example, D:\ARCHIVE) as shown in Figure 19.

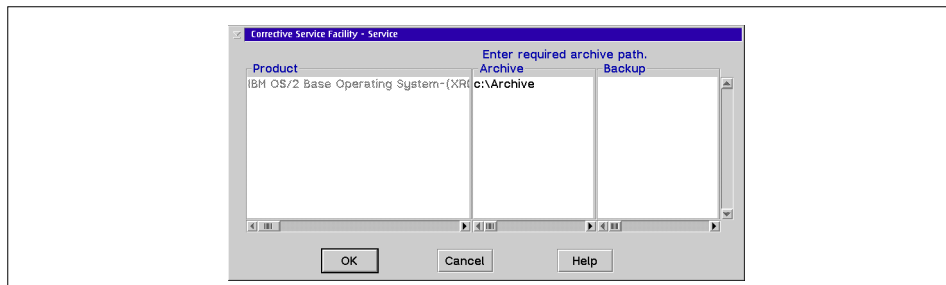


Figure 19. The Files are Archived or Backed Up before Servicing

Note: This ARCHIVE directory is not related to the \OS2\ARCHIVES directory built into OS/2 Warp 4. We recommend specifying a different path. You should specify a different ARCHIVE directory for each product to be serviced.

If the OS/2 FixPak is not the first one being installed, you need to enter the path to the BACKUP directory where a copy of replaced files will be stored (for example, D:\BACKUP).

Note: You must specify a different BACKUP directory for each product to be serviced.

Note

It can take a considerable length of time for the Corrective Service Facility to scan your hard disk for serviceable files. In some cases, it has taken as long as 40 minutes. Please be patient and allow this process to complete.

After entering the paths and clicking **OK** the service starts. During the installation process, you might see messages about Archiving, Updating and Deferring service:

- Archiving saves a compressed copy of the original file in the ARCHIVE or BACKUP path you specified.
- Updating means the original files are replaced with the new ones from this FixPak.
- Deferring service means the file to be updated is currently in use by the system and cannot be updated. The new files from the FixPak are placed in unpacked format in the \IBMCSFLK\FIX directory on the drive with the most free space. They are processed by the locked file device driver during reboot after you shut down the system.

Select **NO** for a redisplay of the "Product List" after the first part of the FixPak application process has completed if this message is displayed. Click **Cancel** and **Exit** to close the Corrective Service Facility window if necessary. Shut down and reboot your system to process the locked files and complete the FixPak installation.

Boot Right After the FixPak Installation

Install only one FixPak at the time. If you run multiple FixPak installation in a row, the locked file device driver may not work correctly and some files may not be processed.

3.7.2 The FSERVICE Command

The Corrective Service Facility (CSF) provides a program, FSERVICE.EXE, for the distribution of maintenance. As mentioned above, this file is provided on one of the kicker diskettes that need to be obtained separately from FixPaks.

FSERVICE is an application similar to RSPINST, supporting input from a response file, and can read the Service Pak files from a redirected drive, which removes the need to insert diskettes.

The following command line parameters are valid for FSERVICE.EXE:

- /R: <Response file>* This specifies the fully qualified path and name of the response file. This parameter is mandatory.
- /S: <Source directory>* This parameter is optional. It specifies the base directory of the ServicePak images. The images must have been prepared prior to service (see Servicing of OS/2 Products). This parameter will override the :SOURCEPATH tag in the response file if the tag exists. No blank spaces are allowed between the colon and the parameters specified for the source directory.
- /SF: <Is source directory on a removable media?>*
Indicates the type of source directory.
Values:
0 = removable (user will be prompted for diskette changes)
1 = non-removable (source directory contains all files and directories delivered with the CSD)
- /T: <Target directory>* It specifies the directory from which the ServicePak will be installed. This parameter is optional if the Service Pak installation is started from a diskette. If the installation is started from a diskette with this parameter specified, the value is not verified.
This parameter is required if the ServicePak installation is started from the hard disk under the OS/2 maintenance system created by SEMAINT. In this case, the value specified in the /T: parameter for FSERVICE should be the same as for SEMAINT (for example, C:\SERVICE). No blank spaces are allowed between the colon and the parameters specified for the source directory.
- /L: <Log file>* This parameter is optional. It specifies the fully qualified path and the name of the log file. This parameter overrides the :LOGFILE tag in the response file, if the tag exists.
If no log file is specified,

OS2\INSTALL\SERVICE.LOG will be used. No blank spaces are allowed between the colon and the parameters specified for the log file.

/CID <System booted from SEMAINT environment>

This parameter is mandatory if SEMAINT is used. It specifies that a client workstation is serviced using SEMAINT.EXE. If it is booted from diskettes, this parameter must be omitted.

? <Display help panel>

This parameter causes FSERVICE to display help information.

Note

Command line parameters override response file parameters.

3.7.3 Sample ServicePak/FixPak Response File

The following response file allows FSERVICE to perform a default ServicePak/FixPak installation. All files from the ServicePak/FixPak will be applied to the system regardless of date/time stamp and file attributes of files on the system being serviced.

```
*Indicates this is a service
:SERVICE
*Indicates that all version on all partitions will be services
:SYSLEVEL
*Indicates the archive path
:ARCHIVE C:\ARCHIVE
*Indicates to update all files and exit
:FLAGS REPLACE_NEWER REPLACE_PROTECTED EXIT_WHEN_DONE
```

Figure 20. Base Service Response File

3.7.4 Enhanced ServicePak/FixPak Response Files

In regards to the CLIFI (Command Line Interface Feature Installer), all ServicePaks and FixPaks for OS/2 and OS/2-related products must comply with the rules of Feature Installer (FI) and ASD (see "ASD - Alternative Software Delivery" on page 154 for more detailed information). To make sure that FI and Corrective Service Facility work together properly, for example to archive and back out lower levels of a feature or files being serviced, new response files had to be created. The following response files are applicable to different service scenarios.

3.7.4.1 Service Response File Applicable on First Execution

The following is a sample response file that can be used when applying service for the first time using `FSERVICE`, when there is no existing archive of the product being serviced. It will service all partitions and place an archive in each partition. It does not take a backup of changed files.

```
:LOGFILE C:\OS2\INSTALL\SERVICE.LOG
:FLAGS REPLACE_PROTECTED REPLACE_NEWER
:SOURCE A:\
:SERVICE
:SYSLEVEL \OS2\INSTALL\SYSLEVEL.OS2
:ARCHIVE \ARCHIVE
:SERVICE
:SYSLEVEL \MMOS2\INSTALL\SYSLEVEL.MPM
:ARCHIVE \ARCHIVE
*
* End of sample SERVICE response file without backup.
```

Figure 21. Service Response File to Support First-Time-Invocation

3.7.4.2 Service Response File to Support Existing Archive

The following is a sample response file to be used when applying service using `FSERVICE` when there is an existing archive of the product being serviced. This demonstrates the ability to take a backup of changed files.

```
:LOGFILE C:\OS2\INSTALL\SERVICE.LOG
:FLAGS REPLACE_PROTECTED REPLACE_NEWER
:SOURCE A:\
:SERVICE
:SYSLEVEL C:\OS2\INSTALL\SYSLEVEL.OS2
:ARCHIVE C:\ARCHIVE
:BACKUP C:\BACKUP
:SERVICE
:SYSLEVEL C:\MMOS2\INSTALL\SYSLEVEL.MPM
:ARCHIVE C:\ARCHIVE
:BACKUP C:\BACKUPM
*
* End of sample SERVICE response file with backup.
```

Figure 22. Service Response File to Support Existing Archive

3.7.4.3 Service Response File to Support Backing Out to an Archive

The following is a sample response file to be used when backing out to the

archive level of a product.

```
:LOGFILE C:\OS2\INSTALL\SERVICE.LOG
:TARGET ARCHIVE
:BACKOUT
:SYSLEVEL C:\OS2\INSTALL\SYSLEVEL.OS2
:BACKOUT
:SYSLEVEL C:\MMOS2\INSTALL\SYSLEVEL.MPM
*
* End of sample BACKOUT to archive response file.
```

Figure 23. Service Response File to Support Backing Out an Archive

Note: You may need to leave an uncommented blank line after the last :SYSLEVEL line when backing out.

3.7.4.4 Service Response File to Support Backing Out to a Backup

The following is a sample response file to be used when backing out to the backup level of a product.

```
:LOGFILE C:\OS2\INSTALL\SERVICE.LOG
:TARGET BACKUP
:BACKOUT
:SYSLEVEL C:\OS2\INSTALL\SYSLEVEL.OS2
:BACKOUT
:SYSLEVEL C:\MMOS2\INSTALL\SYSLEVEL.MPM
*
* End of sample BACKOUT to backup response file.
```

Figure 24. Service Response File to Support Backing Out to a Backup

3.7.4.5 Service Response File to Support Committing

The following is a sample response file to be used when committing a product.

```
:LOGFILE C:\OS2\INSTALL\SERVICE.LOG
:COMMIT
:SYSLEVEL C:\OS2\INSTALL\SYSLEVEL.OS2
:COMMIT
:SYSLEVEL C:\MMOS2\INSTALL\SYSLEVEL.MPM
*
*End of sample COMMIT response file.
```

Figure 25. Service Response File to Support Committing

3.7.4.6 Service Response File to Support Archive Redirection

The following is a sample response file to be used when redirecting an archive of a product to another existing archive location. One example of this would be for using a shared network archive. Note that the archive directory specifies the location of an existing archive to which the current product is being redirected. In this example, the arbitrary drive shows *s:*, which may be a LAN drive.

```
:LOGFILE C:\OS2\INSTALL\SERVICE.LOG
:REDIRECT
:SYSLEVEL C:\OS2\INSTALL\SYSLEVEL.OS2
:ARCHIVE S:\ARCHIVE
:REDIRECT
:SYSLEVEL C:\MMOS2\INSTALL\SYSLEVEL.MPM
:ARCHIVE S:\ARCHIVEM
*
* End of sample REDIRECT response file.
```

Figure 26. Service Response File to Support Archive Redirection

3.7.5 Logging Information

The CSF program will log information pertaining to the service being applied. This information includes:

- Components serviced
- Date of service
- Directories serviced
- Files serviced

Unless otherwise specified in the CSF response file `:LOGFILE` tag, the log file will be named `SERVICE.LOG` and will reside in `\OS2\INSTALL` directory. If the file already exists, then logging information will be appended.

3.7.6 Interrupted Service

If the process is interrupted (after a power failure for example), it can simply be restarted by rebooting the system and going through the installation process again.

Files already updated will not be replaced again due to the checking process performed by the ServicePak (as explained in “The FSERVICE Command” on page 65).

3.7.7 CSF Response File Keywords

The response file required by CSF (Corrective Service Facility) should not be confused with the response file used by the installation process. This response file is a flat ASCII file consisting of tags and parameters.

The asterisk in the first column marks a comment line. A default response file should be provided on the ServicePak diskettes along with a README file that explains the usage of the response file in detail. As the invocation and the keywords changed in the past, we recommend checking all information coming with the ServicePak to find out if there is anything new.

There are several ways to create a valid ServicePak response file:

- Use the default response file provided on of the ServicePak diskettes.
- Create a file with an ASCII editor using the keywords specified below.
- Place the ServicePak diskette 1 in drive A: and execute `A:\SERVICE`. Using the PM interface, select the subdirectories that should be serviced. Close the window. A file called `CSF$_SEL.000` has been created in the root directory, which is a valid ServicePak response file for `FSERVICE.EXE`.

The following list shows the valid response file tags and their purposes:

<code>:SERVICE</code>	Indicates this to be a service. In other words, the <code>:SERVICE</code> tag will install the necessary maintenance to the operating system.
<code>:SYSLEVEL</code>	Indicates the syslevel files that should be serviced. If no parameter follows the <code>:SYSLEVEL</code> tag, all partitions will be serviced. That means that by entering a fully qualified path for a syslevel file, you can include partitions or product modules in the servicing, and by not mentioning them, you can exclude them. This keyword is mandatory and must follow the <code>:SERVICE</code> keyword.

:ARCHIVE This keyword is followed by the fully qualified path for an archive directory for the product that is serviced. In most CSD installs, this parameter is mandatory for a successful install.

:BACKUP This keyword is followed by the fully qualified path for a backup directory for the product that is serviced if an archive for the product was already used.

:BACKOUT This keyword is used to recover the installed system to the previous level. It must be followed by the **:SYSLEVEL** tag that specifies the related syslevel file.

:REDIRECT This keyword redirects the **FSERVICE** routine to another directory than the current directory. It must be followed by the **:SYSLEVEL** tag that specifies the related syslevel file and by the **:ARCHIVE** tag.

:COMMIT This keyword commits a product to a specific ServicePak. That means that there is no **BACKOUT** possible.

:LOGFILE <path\filename>

Specifies the log file. All logged information will be appended to this file with a time stamp as the first entry. The file will be created if it does not already exist. This tag will be overridden by the **/L:** command-line parameter in the **FSERVICE** statement if it is specified.

:FLAGS <flag1> <flag2>

This optional tag specifies optional flags. The following are the flag options that can be used:

REPLACE_NEWER	Replace files that have dates later than the corresponding file on the CSD. If this is not specified, the user is prompted if any newer files are found.
REPLACE_PROTECTED	Replace files that are read-only, hidden or system files. If this is not specified, the user is prompted if any protected files are found.
EXIT_WHEN_DONE	Specifies that FSERVICE should exit when the maintenance process is completed.

:SOURCE <path\filename>

Specifies the source file. If a source was specified in the invocation of **FSERVICE**, the one specified in the response file will be overridden.

:TARGET <path\filename>

Specifies the target for the :BACKOUT parameter. If :BACKOUT is used, the :TARGET parameter is mandatory.

Also check Chapter 9.2.5, "Other Information on the Code Server" on page 316 for more information about FixPaks and ServicePaks.

Chapter 4. Utilities to Ease Remote Installation

This chapter illustrates the most common utilities used for remote installation of OS/2 Warp 4, such as:

- GETOSCID
- SEDISK
- SEIMAGE
- SEINST
- RSPINST
- DSPINSTL
- GETBOOT
- GETREXX
- THINSRV
- THINIFS
- IFSDDEL
- CASINSTL
- CASAGENT
- CASCKREX
- CASDELET
- NVDMBDSK
- CLNDESK
- CLIFI / FI
- RINSTPRN
- RINSTCFG

Not all typical utilities used in the CID process are illustrated here. Find more information about utilities and their invocation syntax in Chapter 10.9, "Creating Change Files" on page 361.

In addition, CID and utility-specific return codes are discussed at each segment of this chapter.

Important Remarks on the Code Server

For scenarios covered in this redbook, the CID utilities that come with OS/2 Warp 4 are copied to the \CID\IMG\OS2WARP4 image directory to better reflect which tools belong to which component.

If you are upgrading your code server using the OS/2 Corrective Service Facility (CSF) either through the Internet, Intranet, FixPak diskettes or remotely through a software distribution manager, make sure CSF does not update the image tree. However, after the FixPak has been applied you may need to update the CID utilities manually. Always read the README file that comes with all ServicePaks/FixPaks.

Throughout this chapter, it is assumed that D: is the code server's drive, and E: is the CD-ROM drive letter.

4.1 Loading OS/2 CID Utilities to the Code Server

Four programs are shipped with the product which are must be unpacked for successful CID installation of OS/2. These programs are packaged in a bundle file named CID shipped on Diskette 7. On the CD-ROM, the CID macro file is stored on \OS2\IMAGE\DISK_7. The programs are:

- SEIMAGE.EXE
- SEDISK.EXE
- SEMAINT.EXE
- SEINST.EXE.

If you plan to distribute future versions of OS/2, look for a README.CID file on any of the first couple of diskettes. This file includes the latest changes and information if anything has changed for that version.

The `UNPACK` command is needed to get the utilities out of the bundle files. The `UNPACK` programs are also version dependent. If you are at the same level of OS/2 on the code server as the one you wish to remote install, you do not need to copy the unpack commands from diskette. They are in the \OS2 directory already and accesible anywhere through the `PATH` statement.

The `SEINST.EXE` program calls `RSPINST.EXE` to do the actual response file driven OS/2 installation. The `RSPINST.EXE` file can be found in the bundle file `REQUIRED` on OS/2 Diskette 7 or in \OS2\IMAGE\DISK_7 on the CD-ROM. A sample response file for OS/2 Warp 4 (`SAMPLE.RSP`) is also in

bundle file REQUIRED on Diskette 7. The SAMPLE.RSP must be modified before you can use it as a base OS/2 response file for remote distribution.

SHPIINST.DLL is needed if you want to install printers using the RINSTPRN program that is shipped with this redbook. For more information, see Chapter 4.15.2, "The RINSPRN Command" on page 131. For OS/2 Warp 4, SHPIINST.DLL is in the bundle file named BUNDLE on Diskette 2 or in the \OS2IMAGE\DISK_2 on CD-ROM.

Loading OS/2 CID utilities can be done as described below or by using the LCU Utility GETOSCID (Chapter 4.3, "The GETOSCID Command" on page 78). The DLL files mentioned will be unpacked if GETREXX is used later during the setup of the code server.

If you are using NetView DM/2 as your software distribution server, you probably will not run GETREXX. Refer to the corresponding chapter how to enable REXX support in a NetView DM/2 environment.

4.2 Unpacking the OS/2 Warp 4 CID Utilities

Change to the subdirectory on the codeserver where you store the image files for OS/2 Warp 4.

The CID and REQUIRED bundles are on OS/2 Warp 4 Diskette 7. Since OS/2 Warp 4 is only shipped on CD-ROM, the following assumes that you have the CD-ROM available. Change to the CD-ROM's \OS2IMAGE\DISK_7 directory and execute the following commands from an OS/2 window:

```
UNPACK E:\OS2IMAGE\DISK_7\CID          D:\CID\IMG\OS2WARP4
UNPACK E:\OS2IMAGE\DISK_7\REQUIRED    D:\CID\IMG\OS2WARP4 /N:RSPINST.EXE
UNPACK E:\OS2IMAGE\DISK_7\REQUIRED    D:\CID\RSP\OS2WARP4 /N:SAMPLE.RSP
COPY   E:\OS2IMAGE\PMDD_1\PRDESC.LST  D:\CID\RSP\OS2WARP4
```

Figure 27. Loading Essential OS/2 Warp 4 CID Utilities

The BUNDLE bundle containing SHPIINST.DLL is on OS/2 Warp V 4 Diskette 2, which also includes INSCFG32.DLL.

```
UNPACK E:\OS2IMAGE\DISK_2\BUNDLE D:\CID\DLL /N:INCSFG32.DLL
UNPACK E:\OS2IMAGE\DISK_2\BUNDLE D:\CID\DLL /N:SHPIINST.DLL
```

Figure 28. Loading Required DLLs for Remote Printer Install

And for a successful installation of OS/2 Warp V 4 on an HPFS formatted partition, the driver, UHPFS.DLL, is needed:

```
COPY E:\OS2IMAGE\DISK_2\UHPFS.DLL D:\CID\DLL
```

Figure 29. Necessary DLL for HPFS Partitions

How to Locate a File in a Bundle

The placement of the bundle files, their names and their contents may vary between OS/2 versions. The easiest way to find out where they are and what files reside in them is to proceed as follows:

1. Install or copy the OS/2 images to the code server first. SEIMAGE is required in case of transferring product image diskettes.
2. Change to the OS/2 image directory of the version you need to work with. For OS/2 Warp 4, that would be D:\CID\IMG\OS2WARP4.
3. Change to a "diskette" directory. The diskette directories are named DISK_0, DISK_1 and so forth. The "Installation diskette" (DISK_0) does not contain any bundle files; so you can start on Diskette 1 or DISK_1. Use the /SHOW parameter with the UNPACK command to display the contents without unpacking the bundled files.

```
UNPACK * /SHOW | MORE
```

The MORE command reads data from the standard input device and sends data to the standard output device (usually the display) one full screen at a time. After each screen, the OS/2 operating system pauses with the message --More-- until you press any key to continue.

Repeat this for each disk or subdirectory to find the file you are searching for.

4.3 The GETOSCID Command

GETOSCID assumes that the OS/2 diskette images are already installed on the code server. Look in Chapter 4.1, "Loading OS/2 CID Utilities to the Code Server" on page 76, on how to do it. With the GETOSCID command, you can unpack all CID utilities at once; so if you do not have a very special reason for using GETOSCID, skip this part.

This utility copies the OS/2 CID modules that are required on the code server from the OS/2 diskette images tree. `GETOSCID` gets the following modules and places them in the target directory:

- SEDISK.EXE
- SEIMAGE.EXE
- SEINST.EXE
- SEMAINT.EXE
- RSPINST.EXE
- SAMPLE.RSP
- UNPACK.EXE
- UNPACK2.EXE

GETOSCID Syntax

```
GETOSCID <source> <target>
```

<source> Fully fully qualified path of the OS/2 diskette images tree.

<target> Fully qualified target path where the files should be copied to.

GETOSCID Example

```
D:\CID\EXE\GETOSCID D:\CID\IMG\OS2WARP4 D:\CID\EXE
```

4.4 The SEDISK Command

SEDISK.EXE is a utility that creates the "Installation Diskette" (DISK_0) "Diskette 1" (DISK_1) and "Diskette 2" (DISK_2) of OS/2. Warp 4. The diskettes as created do not have LAN transport drivers or redirector code. These must be added to the boot diskettes afterwards.

SEDISK requires three formatted diskettes and also requires that the diskette images have previously been installed on the hard disk of the code server.

SEDISK Syntax

```
SEDISK /S: /T: [/P]:
```

/S: Fully qualified source path

Fully qualified path to the OS/2 diskette images, which can be on a local hard drive or redirected drive.

/T: Target drive
/P: Optional parameter for PCMCIA support

If you want to create boot diskettes for laptop systems with PCMCIA slots (an IBM ThinkPad for example), use the /P: parameter. The PCMCIA.SYS driver (as well as the appropriate socket driver) will be copied over to the boot drive. The PCMCIA_ID number represents a number associated with the laptop/notebook desired. Look up the keyword PCMCIA in the default response file, SAMPLE.RSP, to figure out what number to specify:

For example: Specify /P:17 for an IBM Thinkpad 750.

SEDISK Example

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A:
```

This command creates the set of three OS/2 Warp 4 boot diskettes using the files in the D:\CID\IMG\OS2WARP4 directory.

NOTE on SEDISK

If the client machine has some certain requirements, like special PCMCIA drivers, you have to add these manually to the boot diskettes.

4.4.1 SEDISK Return Codes

CID0100E	You have entered one or more invalid arguments.
CID0106E	Failed to update %1.
CID0107E	Failed to create boot disk.
CID0108E	Failed to copy files.
CID0111E	The diskette does not contain enough free space.
CID0112E	The diskette must be high density.
CID0115E	This diskette is write protected.
CID0116E	The %1 parameter must be a local diskette drive.
CID0117E	The path %1 is not found.
CID0120E	You have entered an invalid parameter in %1.
CID0121E	The %1 parameter must contain a full path name.
CID0122E	You must enter the %1 and %2 parameters.

CID0124E The %1 and %2 parameters cannot be on the same drive.

4.5 The SEIMAGE Command

SEIMAGE.EXE is a utility used to automate the creation of a OS/2 Warp 4 directory structure on the code server, for use by the installation process. SEIMAGE copies all OS/2 diskettes to a specified target directory. Since OS/2 Warp 4 comes only on CD-ROM, this command is not applicable to OS/2 Warp 4. Instead of using SEIMAGE, you instead copy the CD-ROM's \OS2IMAGE directory onto the code server.

The diskettes are copied into directories with the same name as their volume labels. For example, volume label "DISK 0" will be copied into a DISK_0 subdirectory within the specified target directory. Directories are created by SEIMAGE if they do not exist.

The program will prompt the user to insert diskettes and copy all files from the diskettes to the appropriate directories.

SEIMAGE Syntax

```
SEIMAGE /S: /T:
```

- /S: Specifies the source drive from which the diskettes will be loaded.
- /T: Specifies the fully qualified target path to which the files will be copied. This directory will be shared and become the source path for installation by RSPINST.EXE.

SEIMAGE Example

```
[D:\CID\IMG\OS2WARP4\SEIMAGE /S:A: /T:D:\CID\IMG\OS2WARP4
```

The command displayed above will copy all product diskette images.

The OS/2 Warp 4 CD-ROM has the necessary directory structure, so the easiest way is to XCOPY the \OS2IMAGE directory to the code server's \CID\IMG\OS2WARP4 directory:

XCOPY Os2/Warp from CD ROM

```
XCOPY E:\OS2IMAGE D:\CID\IMG\OS2WARP4 /S /E
```

4.5.1 SEIMAGE Return Codes

The following return codes are produced by SEIMAGE.

CID0051E %1 - You have inserted %2 in drive %3.
CID0053E %1 - was passed an invalid parameter: %2.
CID0054E %1 - was called with an incorrect number of parameters.
CID0055E %1 - Number of parameters entered = %2.
CID0056E %1 - Return code from "%2" = %3.
CID0057E %1 - %2 file could not be read.

4.6 The SEINST Command

SEINST installs the OS/2 base operating system on the remote client workstation. SEINST can be run from a maintenance partition created with the SEMAINT command or from boot diskettes created with the SEDISK command.

SEINST executes RSPINST.EXE, which reads the specified response file and performs the installation. If SEINST is run from a maintenance partition which was created with the SEMAINT command, SEINST also cleans up the directory from which the system was booted.

SEINST Syntax

```
SEINST /S: /T: /B: /L1: /R:
```

- /S: Specifies the fully qualified source path to the OS/2 images. This parameter is required.
- /T: Specifies the fully qualified path to the target directory from which the system was booted. If the system was booted from a maintenance system on the hard disk, this path matches the /T: parameter passed on the previous invocation of the SEMAINT program. This parameter is required if booted from the hard disk, but optional if booted from diskette. If this parameter is specified when booted from diskette, no parameter validation is done on its value.
Note: Because SEINST deletes all files as it cleans up this subdirectory, back up the files you want to save.
- /B: Specifies the target boot drive from which the system boots after installation. The drive must be a local drive. This parameter is required.
- /L1: Specifies the fully qualified name of the log file into which log information is to be placed. The directory in which the log file is to be placed must already exist. This parameter is required.

/R: Specifies the fully qualified file name of the response file supported by RSPINST. This parameter is required.

Notes:

1. Be aware that SEINST does not install all base operating system components such as developer API extensions, also known as Open32, JAVA 1.02 or any BonusPak components. These components are installed through the built-in Feature Installer.
2. SEINST can execute CLIFI (Command Line Interface Feature Installer) as a SEINST install through LCU (LAN CID Utility) does, however, neither NetView DM/2 nor TME10 SD 3.1.3 supports invocation of CLIFI out of SEINST. In these cases, CLIFI has to be executed externally through a change profile.
3. In some cases SEINST may produce return codes, Feature-Installer-related ones, that cannot be handled by NetView DM/2 or TME 10 SD 3.1.3. If you experience something like this we recommend using SEINST.EXE from OS/2 Warp Connect (OS/2 Warp 3 family does not have Feature Installer as part of the operating system and, therefore, cannot produce the return codes mentioned above).

Following is a SEINST example of when the remote workstation was booted from boot diskettes:

SEINST Example 1

```
X:\CID\IMG\OS2WARP4\SEINST /S:X:\CID\IMG\OS2WARP4 /B:C:  
/R:X:\CID\RSP\OS2WARP4\RESPONSE.FIL /T:A:\  
/L1:X:\CID\LOG\OS2WARP4\CLIENT.LOG
```

Following is a SEINST example when the remote workstation was booted from a maintenance partition (in this example, the maintenance partition is the F: drive):

SEINST Example 2

```
X:\CID\IMG\OS2WARP4\SEINST /S:X:\CID\IMG\OS2WARP4 /B:C:  
/R:X:\CID\RSP\OS2WARP4\RESPONSE.FIL  
/L1:X:\CID\LOG\OS2WARP4\CLIENT.LOG /T:F:\
```

Note: For more information about the RSPINST command and the use of response files, look at Chapter 7, “Creating Response Files” on page 181. This section describes how to run installations through response files initiated

from diskette and might be beneficial for understanding an installation through a response file.

4.6.1 SEINST and SEMAINT Return Codes

This is a collection of the ReturnCodes of SEINST and SEMAINT.

CID0001E	%1 - was called with an incorrect number of parameters.
CID0002E	%1 - was passed an invalid parameter = %2.
CID0004E	%1 - %2 could not be executed or terminated abnormally.
CID0005E	%1 - %2, %1 completed with return code 0x%3.
CID0006E	%1 - Number of parameters entered = %2.
CID0007E	%1 - The log file cannot be placed in target directory.
CID0010E	An error occurred while copying %1.
CID0011E	An error occurred while executing %1.
CID0012E	%1 was started with invalid parameters.
CID0013E	The following parameters were invalid: %1 %2.
CID0014E	%1 was started with insufficient parameters.
CID0015E	The following parameters were missing: %1.
CID0018E	An error occurred while updating %1.
CID0019E	%1 ended with errors. Return code = 0x%2.
CID0020E	No %1 files were found.
CID0023E	%1 was not found.
CID0024E	Unable to create the directory %1.
CID0025E	The %1 and %2 parameters cannot be the same.
CID0026E	Cannot open log file %1.
CID0027E	Return code from %1 was %2.
CID0028E	Return code from %1 of %2 was %3.
CID0029E	The %1 and %2 parameters must be on a local drive.
CID0030E	An error occurred while validating %1.
CID0031E	%1 - The %2 and %3 parameters cannot be the same.
CID0032E	%1 - Return code from %2 was %3.

4.6.2 RSPINST Return Codes

Return codes that are returned from RSPINST.EXE were difficult to diagnose since documentation was not available. The following illustrates error messages that may occur when performing an OS/2 installation. These may occur during CID- or non-CID-type installations.

The numbers of the error messages correlate. For example, you may see that a message like RSPINST has a return code of 941. Looking through this list of possible errors, you will find that a 941 means there is a FORMAT error.

The `RSPINST` error messages are documented in the "online" (INF file version) MPTS Configuration Guide; however, the error messages are not in the current version of the hardcopy. They are also documented in the *LAN CID Utility Guide*, S10H-9742, and in the README.CID file on DISK_0 of OS/2 Warp 4.

The following messages are used for LAN install and response file logging, messages, errors, etc.

INS0702 ERROR: invalid line "%1", invalid line in response file.
INS0707 ERROR: Invalid key value "%1".
INS0708 Response file interface is not being used, response file not found.
INS0710 Windows system missing or invalid.
INS0711 Cannot format Windows partition if you support it.
INS0712 Response file keyword conflict.

The Messages 898 - 920 are miscellaneous messages.

INS0901 Partition Size Error
 To Install Operating System/2 you must have a primary partition of at least %1MB
INS0905 FDISK unsuccessful.
INS0906 Less than xMB primary partition exists.
INS0907 Primary partition exists, greater than x1MB available.
INS0908 No primary partition exists, less than xMB available.
INS0909 Greater than xMB primary partition exists.
INS0911 Could not create file %1.
INS0914 System Installation detected an internal error.
INS0915 System Installation failed to initialize.
INS0916 System Installation failed to start the session.
INS0920 Load Module Error
 System Installation failed trying to load a module into memory.
INS0921 Target Drive Error. Use FDISK to add target drive to Boot Menu.

Messages 930 - 959 are error messages.

INS0932 Copy Error
 An error occurred when System Installation tried to copy a file.
INS0933 Delete Error
 An error occurred when System Installation tried to delete a file.
INS0934 Device Configuration Error
 An error occurred when System Installation tried to determine your system configuration.

INS0935	Close Error An error occurred when System Installation tried to close a file.
INS0936	Make Directory Error An error occurred when System Installation tried to create a directory.
INS0937	Rename Error An error occurred when System Installation tried to rename a file.
INS0938	Open File Error An error occurred when System Installation tried to open a file.
INS0939	Read Error An error occurred when System Installation tried to read a file.
INS0940	Write Error An error occurred when System Installation tried to write a file.
INS0941	Format Error An error occurred when System Installation tried to format your fixed disk.
INS0942	Display Error An error occurred when System Installation tried to display a panel.
INS0944	Display Driver Install Error
INS0945	Format Error The target drive is not formatted and FormatPartition (or FormatFAT or FormatHPFS) was not set for the drive.
INS0946	Video System Error System Installation detected a video system error. Check your video adapter and display.
INS0947	System Install Internal Error System Installation detected an internal error.
INS0948	Error accessing OS/2 ini file.
INS0949	System File Transfer Error An error occurred when System Installation tried to transfer system files to your fixed disk. Your fixed disk may be bad.
INS0950	Unpack File Not Found No files matched the passed file specification.
INS0951	Unpack Partial Copy Only some files were copied. You may be out of disk space.
INS0952	Unpack Ctrl+Break Error A Ctrl+Break was detected by Unpack. The program was terminated.
INS0953	Unpack Critical Error A Critical Error occurred during a file decompression or copy.

- INS0954 Execute Program Error
An error occurred while trying to execute a program.
- INS0955 Get/Set file Attributes Error
An error occurred while trying to get or set the attributes of a file.
- INS0957 Memory Allocation Error
An error occurred when System Installation tried to allocate a segment of memory.

The range of messages 960 - 979 are used for logging information to the system installation log file.

- INS0961 %1 copy is complete.
- INS0962 Formatting fixed disk.
- INS0963 Installation is complete.
- INS0964 Model = %1.
- INS0965 Renaming files %1.
- INS0966 %1 is being copied to your fixed disk.
- INS0967 System files are being copied to your fixed disk.
- INS0968 System file transfer is complete.
- INS0969 Copying files %1.
- INS0971 Format of fixed disk is complete.
- INS0973 Making directory %1.
- INS0974 Deleting file %1.
- INS0975 The Hardware Systems Programs Diskette does not have all the necessary files.
- INS0976 Installing %1.
- INS0979 Submodel = %1.

The range of messages from 980-989 deals with Dual Boot support.

- INS0980 No Dual Boot installed.
- INS0981 Dual Boot installed.
- INS0982 Dual Boot Installation Warning
OS/2 Installation is unable to completely install the Dual Boot feature because it could not find a DOS CONFIG.SYS file.
- INS0983 Dual Boot Installation Warning
OS/2 Installation is unable to completely install the Dual Boot feature because the SHELL statement in the DOS CONFIG.SYS file is incorrect or missing.
- INS0984 Dual Boot Installation Warning
OS/2 Installation is unable to install the Dual Boot feature.
- INS0985 Dual Boot Installation Warning
OS/2 Installation is unable to install the Dual Boot feature because the DOS version you are using is not DOS version 3.2 (or a later version).

- INS0986 Dual Boot Installation Warning
OS/2 Installation is unable to install the Dual Boot feature because it could not find the DOS system files.
- INS0987 Dual Boot Installation Warning
OS/2 Installation is unable to completely install the Dual Boot feature because the SHELL statement in the DOS CONFIG.SYS file is incorrect. This statement indicates the DOS COMMAND.COM file resides in the root directory of drive C.
- INS0988 Dual Boot Installation Warning
OS/2 Installation is unable to completely install the Dual Boot feature because it could not find the DOS COMMAND.COM file in the directory specified in the SHELL statement in the DOS CONFIG.SYS file.
- INS0989 Dual Boot Installation Warning
OS/2 Installation is unable to completely install the Dual Boot feature because it could not validate the SHELL statement in the DOS CONFIG.SYS file.

The range of messages from 1000 on is for system errors.

INS1000 - INS1020

- INS1000 System Installation detected an internal error (00-20).
- INS1021 Invalid Path
The path is not correct. Retype the entry.
- INS1022 Invalid Filename
The filename is not correct. Retype the entry.
- INS1023 Number Out of Range.
The number specified is not correct. Retypenumber between %1 and %2.
- INS1024 System Installation detected an internal error(24).
- INS1025 No Data Entry
An entry must be made in this field before the program can continue.
- INS1026 System Installation detected an internal error(26).
- INS1060 Invalid Base Product Level, incorrect version.
- INS1061 Invalid Base Product Level, incorrect type.
- INS1062 Invalid Base Product Level, missing SYSLEVEL file.
- INS1063 Memory Allocation Error
- INS1064 CheckSum Failure, unable to OPEN or READ specified file.
- INS1065 CheckSum Failure, unknown CheckSum return code.
- INS1066 Invalid Base Product Level
- INS1067 Invalid File System
- INS1068 Microsoft Windows NT Files Found

INS1069 The version of OS/2 on the target file system is newer than the OS/2 Installation files.

4.7 Display Driver Install

You can use the `DSPINSTL` command to install a display driver that suits your environment with a specified screen resolution. The installation must be invoked after you have finished the OS/2 base operating system installation because you need to boot into a PM-based environment.

The `DSPINSTL.EXE` program can be found in the `\OS2\INSTALL` directory at the local workstation.

DSPINSTL Syntax

```
DSPINSTL /PD: /S: /T: /RES: /U
```

<code>/PD: <dsc_file></code>	The fully qualified .DSC file name. Must fit the display adapter in the workstation.
<code>/S: <source_path></code>	The fully qualified path to the OS/2 images.
<code>/T: <target_drive></code>	The target drive or bootdrive.
<code>/RES: <resolution></code>	The resolution to come up in after reboot.
<code>/U</code>	Indicates unattended installation. Must be used for CID Installation.

If you want to install a display driver that is not shipped with the OS/2 package, copy the files related to the driver on the code server. You have to use the path to the files with the `/S:` parameter. Sometimes the vendor of the display adapter provides a tool to install the driver; you can then use the `DSPINSTL` command to set the correct resolution for the display driver after the installation.

Note

If a resolution was passed in that is not supported in the .PMI file, an error will occur.

If a resolution was passed in that is supported in the .PMI file and not supported by the driver, the driver should default to low resolution.

Following is an example how to install an S3 Chip 86C805 at resolution 1024x768x256 through a CID process:

```
DSPINSTL /PD:C:\OS2\INSTALL\PSS3.DSC /S:X:\CID\IMG\OS2WARP4 /T:C:  
/RES:1024x768x256 /U
```

The following figure illustrates .DSC files and their corresponding chip sets.

```

pshead.dsc          "Headland Technology** HT209"
    VIDEO7_HT205_CHIP = 1
    VIDEO7_HT208_CHIP = 2
    VIDEO7_HT209_CHIP = 3
pstrid.dsc         "Trident** Microsystems TVGA8900c"
    TRIDENT_8800_CHIP = 1
    TRIDENT_8900_CHIP = 2
pstseng.dsc        "Tseng** Laboratories ET4000"
    TSENG_ET3000_CHIP = 1
    TSENG_ET4000_CHIP = 2
tliw32.dsc         "Tseng Laboratories ET4000/W32, /W32i, /W32p"
    TSENG_ET4000W32_CHIP = 3
    TSENG_ET4000W32I_CHIP = 4
    TSENG_ET4000W32IB_CHIP = 5
    TSENG_ET4000W32IC_CHIP = 6
    TSENG_ET4000W32PA_CHIP = 7
    TSENG_ET4000W32PB_CHIP = 8
    TSENG_ET4000W32PC_CHIP = 9
pswd.dsc           "Western Digital** WD90C11, C30, C31 (C30 mode only)"
    WESTERNDIG_PVGA1A_CHIP = 1
    WESTERNDIG_WD9000_CHIP = 2
    WESTERNDIG_WD9011_CHIP = 3
    WESTERNDIG_WD9030_CHIP = 4
    WESTERNDIG_WD9026_CHIP = 5
    WESTERNDIG_WD9027_CHIP = 6
pswdc31.dsc        "Western Digital 90C31"
    WESTERNDIG_WD9031_CHIP = 7
pswdc24.dsc        "Western Digital 90C24"
    WESTERNDIG_WD9024_CHIP = 8
wdc33.dsc          "Western Digital 90C33"
    WESTERNDIG_WD9033_CHIP = 9
psati.dsc          "ATI** Mach8, ATI 28800"
    ATI_18800_CHIP = 1
    ATI_28800_CHIP = 2
    ATI_38800_CHIP = 3          8514 CHIP not SVGA
atim32.dsc         "ATI Mach32"
    ATI_68800_CHIP = 4
atim64.dsc         "ATI Mach64"
    ATI_88800_CHIP = 5
psspdw.dsc         "IBM VGA 256c "
    IBM_SVGA_CHIP = 1

```

Figure 30. .DSC Files and the Corresponding Chip Sets (Part 1 of 2)

```

pscl.dsc          "Cirrus Logic** 5422, 5424"
    CIRRUS_5420_CHIP = 1
    CIRRUS_5422_CHIP = 2
    CIRRUS_5424_CHIP = 3
cl54x.dsc        "Cirrus Logic 5426, 5428, 5430, 5434"
    CIRRUS_5426_CHIP = 4
    CIRRUS_5428_CHIP = 5
    CIRRUS_5429_CHIP = 6
    CIRRUS_543X_CHIP = 7
    CIRRUS_5434_CHIP = 8
pss3.dsc         "S3 86C801, 86C805, 86C928"
    S3_86C805_CHIP = 1
    S3_86C928_CHIP = 2
s3864.dsc        "S3 864"
s38641m.dsc      "S3 864 (16M colors with 1MB of Video Mem)"
    #define S3_86C864_CHIP = 4
    #define S3_86C964_CHIP = 5
wp9000.dsc       "Weitek** Power 9000"
    WEITEK_P9000_CHIP = 1
wp9100.dsc       "Weitek Power 9100"
    WEITEK_W5186_CHIP = 2
    WEITEK_W5286_CHIP = 3
    WEITEK_P9100_CHIP = 4

```

Figure 31. .DSC Files and the Corresponding Chip Sets (Part 2 of 2)

Resolutions supported by each driver depend on factors such as:

1. Amount of video memory
2. Resolutions supported by driver

More `DSPINSTL` examples along with the `VCFGCID` command can be found in Figure 97 on page 374, Figure 98 on page 374, and Figure 99 on page 375.

4.7.1 Matrox Millenium Driver Installation for IBM PC 350

The display adapter support provided by OS/2 Warp 4 is quite good; however, it is not complete. Many vendors provide their own display drivers for their display adapter hardware. For example, to operate an IBM PC 350 6887-86B model equipped with a Matrox Millenium graphics accelerator adapter correctly, the drivers provided by the vendor should be used to gain better screen resolution, screen refresh rate and color depth than with the standard VGA. The drivers can be downloaded from the following Web site:

<http://www.matrox.com/mgaweb/drivers/drivers.htm>

At the time this redbook was written, the driver version was 2.13, and the ZIP file name was 1696_213.ZIP. The information provided here may not be valid with different driver versions. Please read the information file included with the driver package when in doubt. The Matrox Millenium driver package is included on the CD-ROM distributed with this redbook. The driver ZIP file is located in the directory \DRIVERS\MATROX and the extracted files are in the directory \DRIVERS\MATROX\DISK1.

Note

Always make sure that the standard VGA driver is installed before installing the Matrox display adapter driver. If other than VGA driver is installed, the Matrox installation may fail.

To install the Matrox Millenium driver, use the UNZIP or PKUNZIP2 command to extract the package. The files included in the ZIP file are:

FIXAUTO.CMD
INSTALL.CMD
MGAX64.DSC
MGAX64.DSP
MGAX64.OS2
MGAX64W.OS2
README.CID
README.OS2
UNINSTAL.CMD

The INSTALL.CMD command file is used to install the driver on the system.

INSTALL.CMD Syntax

```
INSTALL <source path> [/U]
```

Where <source path> is the path where the display drivers reside. If the /U parameter is defined, the installation is run in the unattended mode.

The command can be run without parameters.

The driver is run to the VGA resolution mode as default. If you want to predefine any other resolution and color depth, you must edit the MGAX64.DSP file manually. Modify the file as shown below:

```
:TITLE  
MGAX64
```

```
:KEY
MGAX64
```

```
*****
* PRIMARY SETUP *
*****
```

```
*****
* The following section sets the screen resolution to 1024x768x256 *
*****
:SET_RESOLUTION
WIDTH=1024
HEIGHT=768
COLORS=256
PLANES=1
```

```
*****
* Everything from this point on is default *
*****
```

```
:FILES :MODE=PRIMARY
MGAX64.OS2      %BOOTDRIVE%:
README.OS2     %BOOTDRIVE%:\MGA\OS2
FIXAUTO.CMD    %BOOTDRIVE%:\MGA\OS2
UNINSTAL.CMD  %BOOTDRIVE%:\MGA\OS2
*VVGA.SYS     %BOOTDRIVE%:\MGA\OS2
*VVGA.SYS     %BOOTDRIVE%:\OS2\MDOS
```

The valid screen modes for the Matrox Millenium adapter are as listed below:

- 640x800
- 800x600
- 1024x768
- 1152x864
- 1280x1024
- 1600x1200
- 1600x1280
- 1800x1440

Color depths supported are listed below:

- 256
- 65 536
- 16 777 216

Note

Depending on the type of the board installed, and on how much VRAM is available, not all the resolutions are supported. If an invalid resolution is specified, the driver will boot using the default VGA resolution.

After the display driver install is completed, a new icon is created on the desktop. The icon opens the Matrox display driver settings. Reboot the system after installing the driver before changing any settings.

Double-click on the **MGA Settings** icon to set resolution or change, for example, the WIN-OS/2 font size. The first configuration page and the program icon are shown in Figure 32. Reboot the system after making changes.

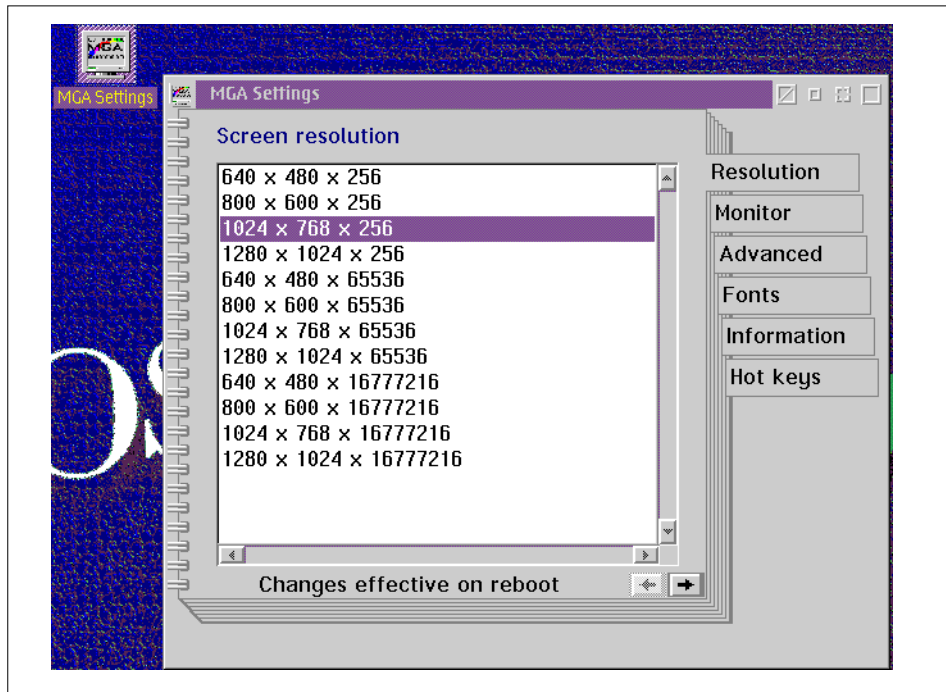


Figure 32. The Matrox Millenium Configuration Program

4.8 Necessary Utilities for LAN CID Utility (LCU)

The following section illustrates utilities you need to get and run in order to use the LAN CID Utility. For example, to have LCU reboot a workstation, the code server must provide the `SETBOOT` command. Or, in order to run the REXX-based LCU, the code server must store all files necessary to run REXX commands at the target workstation being installed remotely.

4.8.1 The GETBOOT Command

In order to complete the installation process, LCU must be able to reboot the client workstation when necessary or appropriate. To do this, LCU uses the `SETBOOT` command available in OS/2.

Since the code server does not necessarily have to be at the same level of operating system as the OS/2 level being installed on the client workstations, we do not want to access the `SETBOOT` module that is in use on the code server.

During the installation, the correct version of `XCOPY.EXE` is needed as well.

You can find the `GETBOOT` utility in the `\CID\LOCINSTU` directory on the OS/2 Warp 4 CD-ROM. `GETBOOT` unpacks `SETBOOT.EXE` and `XCOPY.EXE` files from the OS/2 image tree to the code server.

GETBOOT Syntax

```
GETBOOT <source> <target>
```

<source> Fully qualified source path of the OS/2 diskette images.

<target> Fully qualified target path of the subdirectory where `SETBOOT` command should be copied.

GETBOOT Example

```
E:\CID\LOCINSTU\GETBOOT D:\CID\IMG\OS2WARP4 D:\CID\EXE
```

4.8.2 The GETREXX Command

REXX (Restructured eXtended eXecutor language) is required on the code server to run the LCU REXX command files used to install the requested products remotely at the target workstations. Since the client workstation accesses the LCU command files through a redirected drive, it makes sense

to access the required files for REXX from this particular code server. In this way, the required REXX modules do not have to be on the original boot diskettes.

Since the code server does not necessarily have to be at the same level of OS/2 as the OS/2 level being installed on the client workstations, we do not want to access the REXX modules that are in use on the code server.

GETREXX is a utility that allows the REXX modules to be copied from the OS/2 diskette image tree to the code server. The GETREXX.COM utility unpacks all REXX bundle files, which currently includes:

DLFCLASS.COM
INSCFG32.DLL
OSO001.MSG
REX.MSG
REXH.MSG
REXX.DLL
REXX.IMG
REXX.IR
REXX.TXT
REXXAPI.DLL
REXXC.EXE
REXXCRT.DLL
REXXINIT.DLL
REXXTRY.COM
REXXUTIL.DLL
RXQUEUE.EXE
RXSUBCOM.EXE
SHPIINST.DLL
SOMSEC.DLL
SWITCHRX.COM
UHPFS.DLL
WPCLS.IMP
WPCONST.COM
WPFIND.COM
WPREXX.IMP
WPSINIT.WPS
WPSINST.COM
WPSYSOBJ.COM

For future OS/2 releases, please read all README files with the latest information that come with the OS/2 package. An easy way to find README files is to execute the following command at the code server:

```
DIR D:\CID\IMG\OS2WARP4\README.* /S /P
```

This will find, for example, README.CID and README.INS, both located in \CID\IMGOS2WARP4\DISK_0 directory. Both files should be read before any installation is done.

GETREXX Syntax

```
GETREXX <source> <target>
```

<source> Fully qualified source path of the OS/2 diskette images.

<target> Fully qualified target path of the subdirectory where the REXX files should be copied.

GETREXX Example

```
E:\CID\LLOCINSTU\GETREXX D:\CID\IMG\OS2WARP4 D:\CID\EXE
```

Earlier experiences on code server have shown that sometimes GETREXX (or GETBOOT) fails when called from within another REXX procedure. So please make a habit of checking that the DLL and EXE directories have the expected files; otherwise run GETREXX (or GETBOOT) again.

GETREXX executes the following single-line commands if invoked as displayed in the example above:

```
D:\CID\IMG\OS2WARP4\DISK_3\UNPACK D:\CID\IMG\OS2WARP4\DISK_*\REXX
D:\CID\EXE
D:\CID\IMG\OS2WARP4\DISK_3\UNPACK D:\CID\IMG\OS2WARP4\DISK_*\BUNDLE
D:\CID\EXE /N:OSO001.MSG
D:\CID\IMG\OS2WARP4\DISK_3\UNPACK D:\CID\IMG\OS2WARP4\DISK_*\BUNDLE
D:\CID\EXE /N:INSCFG32.DLL
D:\CID\IMG\OS2WARP4\DISK_3\UNPACK D:\CID\IMG\OS2WARP4\DISK_*\BUNDLE
D:\CID\EXE /N:SHPIINST.DLL
COPY D:\CID\IMG\OS2WARP4\DISK_*\UHPFS.DLL D:\CID\EXE
```

4.9 The SrvIFS Utility

The SrvIFS (Server Installable File System) provides an easy means of redirection. Both functions are provided with this utility: Requester part and Server part. In addition, SrvIFS provides a little security through access list files. However, this utility never reaches OS/2 Warp Server's security.

To work with the different SrvIFS utilities, unpack the zipped SRVIFS file to the code server using the following command:

```
PKUNZIP2 E:\CID\IMG\MPTS\UTILITY\SRVIFS\SRVIFS.ZIP D:\CID\IMG\SRVIFS
```

4.9.1 The THINSRV Command

The `THINSRV` utility is part of the SrvIFS ZIP file. It extracts the necessary mini-server files that can be applied to the code server, verifies provided parameters, copies the necessary files to the target, and updates the `CONFIG.SYS` and `STARTUP.CMD` of the code server workstation to automatically start the mini-server at the code server at system startup.

Following is the syntax of the `THINSRV` utility:

THINSRV Syntax

```
THINSRV /R: [/T:] /S: [/TU:] /U: /L1:
```

/R: Fully qualified path and name of the code server response file.

The content of the response file is equal to the content of the configuration file for `SERVICE.EXE`; so you can use the documentation found in Appendix A.1, “The SrvIFS Utility” on page 441. The response file is copied to the target directory and renamed to `*.INI`.

The `THINSRV` program terminates if required configuration parameters are missing from the INI file. This parameter is required.

/T: Fully qualified target path.

`THINSRV` will create the given directory if it does not exist. If `THINSRV` cannot create the directory for any reason or the path supplied is invalid, `THINSRV` terminates.

If no target path is given, the current boot drive of the system is used. This parameter is optional.

/S: Fully qualified source path.

If this parameter is omitted or invalid, `THINSRV` will terminate.

/TU: Fully qualified path to `CONFIG.SYS`.

`CONFIG.SYS` to be updated with the code server statements by

THINSRV. THINSRV will also create a STARTUP.CMD at this drive if it does not exist already. The statement to start the SrvIFS server is added.

If the /TU: parameter is omitted, the /T: parameter is used as default. THINSRV will terminate if there is no CONFIG.SYS found.

/U: AUTHLIST file source.

The file given by this parameter is use as authentication list file for granting access to the SrvIFS code server.

The file will be copied form source to the target directory specified in the AUTHLIST keyword of the response file. If the target subdirectory does not exists, it will be created. A default AUTHLIST file is located in the same path as the response file.

THINSRV will terminate if the file does not exist.

/L1: Log file name.

The value supplied must be a fully qualified path and file name to the logfile.

THINSRV Example

```
D:\CID\IMG\SRVIFS\THINSRV
/R:D:\CID\RSP\SRVIFS\CIDSRV.INI
/T:C:\SRVIFS
/S:D:\CID\IMG\SRVIFS
/TU:C:\
/U:D:\CID\RSP\SRVIFS\SERVICE.LST
/L1:D>\CID\LOG\SRVIFS\CIDSRV.LOG
```

In this example, all the necessary mini-server files are copied to C:\SRVIFS directory. It will also create a valid CIDSRV.INI based on the response file name specified in the /R parameter. THINSRV validates the content of the response file before creating the actual CIDSRV.INI used by SERVICE.EXE.

The following files are installed on the target by THINSRV:

```
SERVICE.EXE
IFSDEL.EXE
XI1.MSG
XI1H.MSG
```

THINSRV will update the PATH and DPATH statements of the server's CONFIG.SYS file. THINSRV will also add a START statement to the server's STARTUP.CMD file.

4.9.2 The THINIFS Command

THINIFS provides the requester part to THINSRV. It will place all necessary SRVIFS redirection files on the target hard disk or boot diskette. It will also update the CONFIG.SYS to add the required statements for the redirection. This utility works similar to the THINSRV command, as illustrated in Chapter 4.9, "The SrvIFS Utility" on page 98.

The following files are installed on the target by THINIFS:

IFSDEL.EXE
SRVIFS.SYS
SRVIFSC.IFS
SRVATTCH.EXE
XI1.MSG
XI1H.MSG

THINIFS will update the PATH and add CALL, DEVICE, IFS statements to the target's CONFIG.SYS file. It also will generate a SRVATTCH statement in the client's CONFIG.SYS. When the SRVATTCH command is executed using SrvIFS, the client connects to the desired server and gets the redirected drive.

THINIFS can be executed several times if more redirected drives are needed.

THINIFS Syntax

```
THINIFS /S: /T: /SRV: /REQ: /D: [/TU:] [/L1:] [/NS:] [/A:] [/W ]
```

- /S: Fully qualified source path to the SrvIFS utility code on the code server (usually in \CID\IMG\SRVIFS). If omitted or if an illegal value is given THINIFS will terminate.
- /T: Fully qualified target path of the THINIFS installation. If you are creating boot diskettes for the client, this parameter is the drive location where the boot diskettes are located. If you specify a directory, THINIFS attempts to create the target (subdirectory) if it does not exist. Long directory names are supported. THINIFS will terminate if an unsupported drive is supplied.

`/SRV:` NetBIOS name of the code server that is used in the `SRVATTCH` statement. `THINIFS` supports 1-15 alphanumeric characters for the server name. This value specifies the name of the SRVIFS server to which the client should be attached. The client is attached to the default path of the SRVIFS server named. The default path is defined with the `PATH=default_path keyword=value` pair in the SRVIFS server's .INI file, for example: `Path = D:\CID`. In the SRVIFS server's .INI file, the client will be attached to the 'root' of the CID directory structure.

`*P` Prompt
This value presents a prompt where the server name can be entered. It can be entered with the `/SRV` variable.

`\\<server name>\<alias>`
This value should equate to the server name and alias as specified in the code server's .INI file. `THINIFS` will terminate if this parameter is omitted or an invalid/ unsupported value is supplied.

`/REQ:` NetBIOS name of SrvIFS client supplied in the `IFS` statement of the client's `CONFIG.SYS` file. `THINIFS` supports 1-15 alphanumeric characters for the client name. Even though the SRVIFS server and client names are NetBIOS names, SrvIFS does not follow the NetBIOS naming convention.

`*` This value will randomly generate a client NetBIOS name.

`*P` This value prompts for the client name that is to be entered at startup time. It can be entered with the `/REQ` variable.

Note

If you want the user to have a customized prompt for the requested name, modify the `IFS=A:\SRVIFSC.IFS*P` statement generated by `THINIFS` in the `CONFIG.SYS` file. Add the customized prompt in quotes following the `*P` parameter, as follows:

```
IFS=A:\SRVIFSC.IFS *P "Customized Prompt"
```

`*I` This value will attempt to retrieve the NetBIOS name from the `IBMLAN.INI` file, which is the primary configuration file of the OS/2 LAN Server V4/OS/2 Warp Server V4 product. Please read *MPTS Configuration Guide* before using `*I`, since it has some special requirements. `THINIFS` will terminate if this parameter is omitted or an invalid/ unsupported value is supplied.

- /D:** The value supplied is one alphabetic character to be used as the drive letter on the `SRVATTC` statement. The value can be a single alphabetic character (no semicolon) or a single alphabetic character with a colon. `THINIFS` validates the target `CONFIG.SYS` if a `SRVATTC` statement already exists and the drive letter is used. `THINIFS` will terminate if value is not supplied.
- /TU:** The fully qualified path of `CONFIG.SYS`. This parameter is optional. The value supplied is the fully qualified path of the `CONFIG.SYS` that will be updated.
- /NS:** Option for `IFS` statement. This parameter is optional. The parameter indicates the `/S:` flag on the `SRVIFS IFS` statement in the `CONFIG.SYS`. The `/S:` specifies number of NetBIOS sessions. One session is needed per code server the client attaches to concurrently. Valid values are 2 through 9 (default is 5). `THINIFS` will terminate if an unsupported value is supplied.
- /L1:** Log file name. This parameter is optional. The value supplied is fully qualified path and file name of log file. Logging will not occur if this parameter is omitted or is invalid. `THINIFS` will terminate if an invalid/unsupported parameter is provided.
- /A:** Option for `IFS` statement. This parameter is optional. The value is a 0 or 1 (default is 0).
- /W** Option for `IFS` statement. This parameter is optional. This parameter indicates the `/T:` flag on the `SRVIFS IFS` statement in the `CONFIG.SYS`. The `/T:` doubles the NetBIOS timeout value from 15 to 30 seconds. This is useful in environments bridged by lower line speeds.

In the sample LCU command file (`\SG242010\LCU\DEFAULT.CMD`) on the CD-ROM provided with this book, `THINIFS` is called twice. The first time it creates a `SRVATTC` statement that will connect to the code server and gets a redirected drive for the `CID` directory tree with Read/Execute access. (Usually the redirected drive is attached to the `\CID` directory with subdirectories.) It is called a second time to create a `SRVATTC` statement that will get a redirected drive with Read/Write access to enable the client to write log files back to the code server. (Usually the second redirected drive is attached to the `\CID\LOG` directory with subdirectories.)

Example Execution of THINIFS: Base Connect

```
D:\CID\IMG\SRVIFS\THINIFS /S:D:\CID\IMG\SRVIFS /T:A: /SRV:CIDSRV
/REQ:*P /D:X /TU:A:\ /NS:5 /A:0 /W
/L1:L:\CID\LOG\SRVIFS\SRVIFS.LOG
```

At bootup time, the user at the client workstation will be prompted for the client name. When entered, the client connects to the mini-server with the name of CIDSRV. No redirected drives are given at that time. Furthermore, since a `SRVATTCH` statement to the LCU redirector's `CONFIG.SYS` file was added, the client will get a redirected drive X: that points to the default path `D:\CID` on the code server.

See `SERVICE.INI` parameters in "The `SERVICE.INI` Keywords" on page 441 for the definition of the default path. Note that `D:\CID` does not permit writing because it is shared as "read only". See also Figure 34 on page 110.

The requester name function, `/REQ`, of the `IFS` statement in the `CONFIG.SYS` works in three ways:

1. As a name to be checked against the authorization list on the code server
2. As a name to be used for a subdirectory on the PerClient feature of the alias statement in the `SERVICE.INI` file
3. As the name used by `CASAGENT` to find the client-specific LCU command file and response files

`CASAGENT` (a command part of LCU) accepts a new parameter, `/REQ:`, and if this is provided, will use the given name as LCU client name and not the `SRVIFS` name. (`SRVIFS` still uses the name given with `THINIFS`, `/REQ`, as the NetBIOS name for the attachment to the code server.)

Alternatively to the `THINIFS` example shown before, you may want to specify a client name along with a specific server attach statement. In our example, the remote client's workstation name is `CLIENT1`. The command entered must be:

Example Execution of THINIFS: First Server Attachment

```
D:\CID\IMG\SRVIFS\THINIFS /S:D:\CID\IMG\SRVIFS /T:A:
/DRV:\CIDSRV\IMG /REQ:CLIENT1 /D:X
```

It is recommend to use a second invocation of `THINIFS` in order to give client workstations access to a separate LOG drive alias in "read/write" mode for

log files. This will prevent client workstations from accessing and modifying product images, LCU command files or response files stored on the code server.

Example Execution of THINIFS: Second Server Attachment

```
D:\CID\IMG\SRVIFS\THINIFS /S:D:\CID\IMG\SRVIFS /T:A:  
/SRV:\\CIDSRV\LOG /REQ:CLIENT1 /D:L:
```

The example execution of `THINIFS` shown above adds a second `SRVATTCH` statement to the LCU redirector's `CONFIG.SYS` file. This statement points to the LOG share name (alias) `\\CIDSRV\LOG` on the code server and will be accessed by the LCU redirector as drive L:.

SRVATTCH Command Before DEVICE and IFS Statements

`THINIFS` will move `SRVATTCH` statements before `DEVICE` and `IFS` statements. This is acceptable since `IFS` and `DEVICE` statements will be processed before the `CALL` statement.

The following figure displays an extract of the `CONFIG.SYS` file after the two server attach commands have been executed:

```
...  
..  
.  
rem *** Start of ThinLAPS additions ***  
device = lanmsgdd.os2  
device = protman.os2  
device = netbeui.os2  
device = netbios.os2  
device = ibmtok.os2  
run = netbind.exe  
run = lanmsgex.exe  
rem *** Start of ThinIFS additions ***  
CALL=A:\SRVATTCH.EXE X: \\CIDSRV\IMG
```

Figure 33. Statements added to `CONFIG.SYS` by `THINIFS`

See Appendix A.1, “The SrvIFS Utility” on page 441 for more information about SrvIFS-related issues.

4.9.3 The IFSDDEL Command

This utility removes files installed by `THINSRV` and `THINIFS` commands. It removes the `SrvIFS` files and the `SrvIFS` statements from the `CONFIG.SYS` and `STARTUP.CMD` files. There is no support for messages or logging.

IFSDDEL Syntax

```
IFSDDEL /T: /TU: /SD:
```

- /T:** Fully qualified target path. This parameter is optional. Fully qualified path of the `SrvIFS` files which are to be deleted. If omitted, the value of the target will be set to current boot drive.
- /TU:** Fully qualified path. This parameter is optional. The value supplied contains the path to the client `CONFIG.SYS` that will have the `SrvIFS` statements deleted. On the code server it is the path to the `STARTUP.CMD`. If omitted, value from the `/T:` parameter will be used. `IFSDDEL` supports up to three `/TU:` parameters. Multiple `/TU:` parameters are usually used when the LCU is installed on a multiboot machine. If an invalid value is specified the `CONFIG.SYS` file or the `STARTUP.CMD` will not be cleaned up.
- /SD** Optional. This parameter is optional and used only on a code server. This parameter indicates that the code server's files and statements in the `STARTUP.CMD` file will be removed. The removed files are the `SrvIFS` executables and any of the configuration files indicated by the statements in the `STARTUP.CMD` file. `AUHLIST` files that are referenced in those configuration (`.INI`) files will also be deleted. Statements will not be removed from `PATH` or `DPATH` statements in the `CONFIG.SYS` file. `IFSDDEL` does not remove itself from the system.

IFSDDEL Example

```
IFSDDEL /T:C:\ /TU:C:\
```

4.10 LAN CID Utility (LCU)

The actual software distribution manager that works with `SrvIFS` is called LAN CID Utility (LCU), originally code-named `CASTLE`. To work with the different LCU utilities, unpack the zipped LCU file to the code server using the following command:

```
PKUNZIP2 E:\CID\IMG\MPTS\UTILITY\LCU\LCU.ZIP D:\CID\IMG\LCU
```

4.10.1 The CASINSTL Command

This utility installs LAN CID Utility on the LAN transport system diskette or on client workstation hard disk.

As a result of `CASINSTL` execution `CONFIG.SYS` is updated and a `STARTUP.CMD` file is created on the LAN transport system diskette or target hard disk. The `CASAGENT.EXE` is started from `STARTUP.CMD` and run from the code server. One of the parameters for `CASAGENT.EXE` is the path to the LCU command file.

CASINSTL Syntax

```
CASINSTL /TU: /CMD: /D /D: /L1: /L2: /PL: /PA: /PD /REQ: /O
```

- `/TU:` Boot drive. Installation boot drive.
- `/CMD:` Fully qualified path. Fully qualified path to the LCU REXX command procedure to be used. Usually this is `X:\CLIENT` (but it depends on the actual `SRVATTCH` statements in the client's `CONFIG.SYS`). The NetBIOS name of the client will be used as name for the client REXX command file by LCU. For example, if the client NetBIOS name is `CYNTHIA`, LCU will try to find and execute an `CYNTHIA.CMD` in the directory path defined with this parameter.
- `/D` Optional. The default LCU REXX command procedure will be used if a client specific command procedure is not found. It requires a `DEFAULT.CMD` in the directory path given in the `/CMD` parameter.
- `/L1:` Fully qualified path and file name of the log file. The log file will be used by `CASINSTL`, `CASAGENT` and the LCU command file. When both LOG files are used, `CASINSTL` logs only to `/L1` log file.
- `/L2:` Fully qualified path and file name of the log file. The log file will be used by `CASAGENT` and the LCU command file. If both `/L1` and `/L2` are used `CASINSTL` will log to the file defined with `/L1` and `CASAGENT` and the LCU command file will log to the file defined with `/L2`. When only `/L2` is used, `CASAGENT` and the LCU.CMD will log to `/L2` and `CASINSTL` will not log at all. If the LAN CID Utility client name is prompted for or randomly selected either by `CASAGENT` or `SRVIFS`, it is recommended that you use the `SRVIFS_REQ` keyword for the log file name on the `/L2` parameter. This ensures that a unique log file is used for each client installed with these diskettes. The `SRVIFS_REQ` keyword tells LAN CID Utility to replace the `SRVIFS_REQ` keyword in the log file name with the LAN CID Utility client name being generated at run time.

- `/PA:` Optional (but required for boot diskettes). Specifies the fully qualified path (pointing to the code server) to the CASAGENT.EXE and SRVREXX.EXE. Usually, this would be X:\IMG\LCU (but it depends on the actual SRVATTCH statements in the client's CONFIG.SYS). This path is added to the client's LIBPATH= and DPATH= statements.
- `/PL:` Optional. Specifies the values of LIBPATH and DPATH statements to be added to LCU redirector's CONFIG.SYS.
- `/O` Optional. Indicates that this is the first time the CASAGENT program has been called.
- `/D:` Optional. Either `/D`, this parameter, or none is used. Name of the default LCU REXX command procedure will be used if a client-specific command procedure is not found. The file name can be indicated with or without the .CMD extension. It must be the directory path given in the `/CMD` parameter. If you created boot diskettes specifying the `/D` or `/D:` parameter, it is important that you use the same parameter on the CASINSTL command inside the default command file that is to be run. If this is not done, after CASINSTL is run inside the command file and the system is restarted, CASAGENT does not know that it should run a default command file.
- `/PD:` Optional. The redirected drive and path to the workstation LAN CID Utility boot Diskette 1 image on the code server. This parameter is used if you want to be able to remove the LAN CID Utility boot Diskette 1 at the beginning of the first section of the installation instead of waiting until just before the first reboot. CASINSTL does not copy the diskette into this directory. It is up to the administrator to ensure that the directory contains the up-to-date LAN CID Utility boot Diskette 1 files.
- `/REQ:` Optional. The LAN CID Utility client name of the target workstation. This parameter makes it possible to use LCU CASAGENT even if the redirected drives to the code server are not accessed through the use of SRVIFS but by some other server/requester software. The supported values are:
 - `*` An alphanumeric name 1 through 20 bytes long. (The underscore character is also allowed.)
 - `*P` This value specifies to prompt for the client name. This value specifies to allow CASAGENT to randomly select an 8-byte LAN Utility client name. If this selection is chosen, either the `/D` or `/D:` parameter is required on the command line.

Only specify the `/REQ:*` or `/REQ:*P` parameters when creating boot diskettes or when preparing a workstation fixed disk for install.

When `CASINSTL` is run from within an LAN CID Utility command file, it is recommended that the `/REQ` parameter be specified as the client name that was passed into the command file.

For example, this can be done in the following manner in the LCU command file:

```
'/REQ:' client
```

Note: At this time, some programs, such as `FSERVICE` and `SEINST`, do not support long file names for the response files and log files. If you are using LAN CID Utility client names more than 8 bytes long and you are using the LAN CID Utility client name for the response file and log file names, it is important that you test your LAN CID Utility command file before using it in a production environment to determine if the install programs you are using support long file names.

CASINSTL to LAN Transport

```
CASINSTL /TU:A:\ /CMD:X:\CLIENT /D /PA:X:\IMG\LCU;X:\EXE;  
/PL:X:\DLL;X:\IMG\LCU; /L2:L:\LCU\ SRVIFS_REQ.LOG /0
```

Note: If you have lines longer than 255 characters in your `CONFIG.SYS` file, `CASINSTL` will display warnings.

Figure 34 shows the boot diskette's `CONFIG.SYS` file after `CASINSTL` execution:

```

buffers=32
iopl=yes
memman=swap,delayswap
protshell=sysinst1.exe
SET OS2_SHELL=CMD.EXE /K A:\STARTUP.CMD
diskcache=D2,LW
protectonly=yes
libpath=.\;\os2\install;X:\DLL;X:\IMG\LCU;
ifs=hpfs.ifs /c:64
pauseonerror=no
.....
device=\dos.sys
set path=.\;\os2;\os2\system;\os2\install;A:;X:\EXE;X:\IMG\LCU;
set dpath=.\;\os2;\os2\system;\os2\install;A:;X:\DLL;X:\IMG\LCU;
set keys=on
basedev=cmd640x.add
basedev=detne2.sys /p:360
.....
device=\refpart.sys

rem *** Start of ThinLAPS additions ***
device = lanmsgdd.os2
device = protman.os2
device = netbeui.os2
device = netbios.os2
device = IBMTOK.OS2
call = netbind.exe
run = lanmsgex.exe
rem *** End of ThinLAPS additions ***
CALL=A:\SRVATTCH.EXE X: \\CIDSRV\CID
DEVICE=A:\SRVIFS.SYS
IFS=A:\SRVIFSC.IFS *I
CALL=A:\SRVATTCH.EXE L: \\CIDSRV\LOG
RUN=X:\IMG\LCU\SRVREXX.EXE

```

Figure 34. CONFIG.SYS File after CASINSTL

4.10.2 The CASAGENT Command

CASINSTL creates a STARTUP.CMD file on the client's boot drive in which CASAGENT is called with parameters. These parameters were provided when CASINSTL was executed.

CASAGENT Syntax

```
CASAGENT /CMD: /D /D: /REQ: /L1:
```

- /CMD:** Fully qualified path of the redirected drive that contains LCU command files.
- /D** This parameter is optional. If the client's unique LCU command file is not found the **/D** parameter indicates that the default LCU command file will be used. If you use this parameter you need a DEFAULT.CMD file in the directory path set by the **/CMD** parameter.
- /D:** This parameter is optional. Only supported by **CASAGENT**. Either **/D**, **/D:** or none can be used. If the client's unique LCU command file is not found the **/D:** parameter indicates the filename of the default LCU command file that will be used. If you use this parameter you need an LCU command file with the name indicated by this parameter in the directory path set by the **/CMD** parameter.
- /REQ:** Optional. See explanation of valid values for **CASINSTL** (4.10.1, "The **CASINSTL** Command" on page 107). If **/REQ** is not defined the **SrvIFS** redirected client NetBIOS name is used if defined. It is set by the **IFS=SRVIFSC.IFS** name statement in the client's **CONFIG.SYS**.
- /L1:** Log file name. This parameter is optional. Fully qualified path and file name of **CASAGENT**'s log file.

LCU Client's STARTUP.CMD

```
CASAGENT /CMD:X:\CLIENT /D /L1:L:\LCU\SRVIFS_REQ.LOG
```

4.10.3 The CASCKREX Command

CASCKREX is a utility to check that **REXX** is initialized (**RUN=SRVREXX.EXE**) on the client workstation. It returns 0 if **REXX** is initialized and otherwise returns 1.

CASCKREX Syntax

```
CASCKREX CASAGENT
```

The **CASAGENT** parameter is optional (but used by **CASAGENT**). When it is defined no output will be made to the screen. **CASCKREX** could also be used manually from a command prompt.

4.10.4 The CASDELETE Command

Removes the `CASAGENT` command from `STARTUP.CMD` and removes all LCU-specific `CONFIG.SYS` statements enforced by `CASINSTL` execution.

CASDELETE Syntax

```
CASDELETE /TU: /PL: /L1:
```

- /TU:** Boot drive. Target drive to clean up. It can be LAN transport system diskette or more likely your just installed OS/2 system's boot drive. It can be invoked more than once.
- /PL:** `DPATH`, `LIBPATH`. It is optional. Specifies the value of `DPATH` and `LIBPATH` statements to be removed from LCU client's `CONFIG.SYS`.
- /L1:** Log file name. This parameter is optional. Fully qualified path and file name of `CASDELETE`'s log file.

Usually `IFSDEL` and `CASDELETE` are called in the last execution of the LCU command file after all products are installed to the client.

CASDELETE Example

```
CASDELETE /TU:C /PL:X:\DLL;X:\IMG\LCU /L1:Z:\LCU\LOG1.LOG
```

4.11 MPTS Utilities

MPTS provides provides one utility that is in charge of installing network support on boot diskettes or maintenance partitions. This utility is called `THINLAPS` and resides on the first MPTS diskette or in the `\CID\IMG\MPTS` directory of the OS/2 Warp 4 CD-ROM. In comparison to previous versions of MPTS, TCP/IP and NetBIOS over TCP/IP (`TCPBEUI`) is supported so that you do not have to have native NetBIOS present for software distribution.

If you plan to use `TCPBEUI` rather than `NetBEUI`, you need to consider a couple of factors to ensure communications between software distribution servers, and remote clients in particular, when more than one subnet is to be served. The presence of a NetBIOS name server is definitely an advantage.

4.11.1 The THINLAPS Command

This utility creates a seed LAN transport system. `THINLAPS` copies necessary LAN transport files from the source path to the target seed system. The `CONFIG.SYS` file on the system is updated with `DEVICE` and `RUN` statements.

Note

Only the network adapters supplied with MPTS are supported. If you wish to use a network adapter that is not supported by MPTS or if you want to use a newer version of the device driver, you need to update the MACS.ZIP file with the device driver files, Network Information File (NIF) and MSG files of the new adapter. You also might need a customized PROTOCOL.INI file to match the adapter parameter.

A CONFIG.SYS file will be created if one does not exist on the target. THINLAPS will complete execution; however, a warning will be generated.

The default PROTOCOL.INI created on the target drive by THINLAPS will contain instructions for modification if the hardware configuration is nonstandard. The PROTOCOL.INI file will be stored on the third OS/2 boot diskette.

Usually, there are memory addresses that need to be set in a way such that the LAN network adapter will not use the same address(es) as any other adapter in the client machine.

THINLAPS Syntax

```
THINLAPS <source> <target> <NIF file name> <parameters>
```

- <source> Fully qualified source path.
- <target> Target drive for seed system. Accepted value is drive letter with a colon followed by a backslash. The target drive can be diskette or hard file.
- <NIF file name> The name of the .NIF file that corresponds to the network adapter driver that will be stored on the target. See documentation about LAN Network Adapters for mapping of the LAN Transport network adapter drivers and the associated .NIF file names.
- <parameters>
- P: The value supplied is the fully qualified file name of a configuration file that will be copied to the target. When /P: parameter is specified, the PROTOCOL.INI file will be placed on the target instead of the default PROTOCOL.INI file that is generated by THINLAPS. It might be necessary to use /P: and provide the fully qualified path to a working

PROTOCOL.INI if you have added an additional adapter to LAPS.

`/ADD:` Fully qualified path to additional driver. You can specify the location of additional network drivers that are not shipped with MPTS. All necessary files must be in the specified location.

`/TU:` Fully qualified path to CONFIG.SYS. You can specify a location for the CONFIG.SYS you want to update. If the CONFIG.SYS cannot be found in the specified path, you will be prompted to enter drive and path.

`/TCPBEUI` NetBIOS over TCP/IP protocol to be installed

`/TCPIP` TCP/IP protocol to be installed

`/NETBIOS` NetBIOS protocol to be installed. The default is NetBIOS if a network protocol parameter is not specified.

`/DHCP` Retrieve IP address from a DHCP server

`/IP:` IP address nnn.nnn.nnn.nnn. Specify a unique IP address for the target system.

`/PIP: <string>` IP address--The user is prompted to enter an IP address at start time. You can specify a string, which is displayed at start time.

`/NM:` Subnet mask nnn.nnn.nnn.nnn

`/H:` Unique hostname for the target system

`/PH: <string>` Hostname. Prompt the user for hostname at start time. A string can be specified that is displayed at start time.

`/DMN:` Domain scope name

`/NSIP:` IP address of the domain name server

`/RT:` IP address of the router

`/NODE:` TCPBEUI node type. The node type can be: `B`, `H`, or `P`

`/RFCN:` Location of the TCPBEUI names file (RFCNAMES.LST)

`/RFCBC:` Location of the TCPBEUI broadcast names file (RFCBCST.LST)

`/RFCDMN:` Location of the TCPBEUI domain names file

`/NBNS:` IP address of NetBIOS name server. Used with TCPBEUI when node type `H` or `P` is specified

/NBDD: NetBIOS datagram distributor. Used with TCPBEUI when node type `H` or `P` is specified

Note: If the client machine is an ISA-bus machine and you are using IBMTOK.NIF, you may need to edit the PROTOCOL.INI on the LAN transport system diskette you just created. It is better to use `/P` and provide a customized PROTOCOL.INI. This is necessary when the `THINLAPS` command is invoked from a software distribution manager in order to install a seed system, which will be automatically rebooted in order to continue the installation of other products immediately.

THINLAPS Example

```
D:\CID\IMG\MPTS\THINLAPS D:\CID\IMG\MPTS A: IBMTOK.NIF
```

The command above will install a seed LAN transport system on the diskette in diskette drive A: from directory D:\CID\IMG\MPTS using IBMTOK.NIF.

A PROTOCOL.INI is created on the target diskette based on a valid Network Information File (.NIF). The name of the .NIF file is supplied as a parameter. The following figure shows the PROTOCOL.INI created based on the IBMTOK.NIF parameter.

```

[PROTMAN]

    DRIVERVERNAME = PROTMAN$

[NETBEUI_NIF]

    DRIVERVERNAME = NETBEUI$
    BINDINGS = MAC
; If running on an ethernet network, remove the semicolon from the
; "ETHERAND_TYPE =" statement to change the protocol convention from
; IEEE 802.3 to DIX 2.0.
; ETHERAND_TYPE = "D"

[MAC]

    DRIVERVERNAME= IBMTOK$

; Remove the semicolon from the "ADAPTER =" statement when this Token
; Ring adapter should be assigned as the second Token Ring adapter
;     ADAPTER = ALTERNATE
;Remove the semicolon from the "RAM =" statement when the Token Ring
;adapter is an AT-bus adapter. Append the value 0xD800 (for primary)
;or 0xD400 (for alternate) after the equals sign.
;     RAM = 0xD800
;     NETADDRESS = " "

```

Figure 35. PROTOCOL.INI Created by THINLAPS

The default PROTOCOL.INI file created by THINLAPS is only valid all LAN adapters that can initialize IBMTOK.NIF, for example Token Ring adapter for microchannel adapters, ISA Auto 16/4 Token Ring and ISA Turbo Token Ring adapter. Some Token Ring adapters, such as the Token Ring 16/4 II adapter, need the correct value of the shared RAM address space.

The following figure shows the boot diskette CONFIG.SYS file after THINLAPS:

```

buffers=32
iop1=yes
memman=noswap
protshell=sysinst1.exe
set os2_shell=cmd.exe
diskcache=D2,LW
protectonly=yes
libpath=.\;.\os2\dll;
ifs=hpfs.ifs /c:64
pauseonerror=no
codepage=850
devinfo=kbd,us,keyboard.dcp
devinfo=scr,ega,vtbl850.dcp
device=\dos.sys
device=\mouse.sys
set path=.\;.\os2;\os2\system;\os2\install
set dpath=.\;.\os2;\os2\system;\os2\install
set keys=on
basedev=cmd640x.add
basedev=detne2.sys /p:360
basedev=ibmkbd.sys
basedev=ibm1flpy.add
basedev=ibm1s506.add
basedev=ibm2flpy.add
basedev=ibm2adsk.add
basedev=ibm2scsi.add
basedev=ibmint13.i13
basedev=os2dasd.dmd
basedev=xdfloppy.flt
device=\testcfg.sys
device=\refpart.sys
rem *** Start of ThinLAPS additions ***
device = lanmsgdd.os2
device = protman.os2
device = netbeui.os2
device = netbios.os2
device = ibmtok.os2
run = netbind.exe
run = lanmsgex.exe

```

Figure 36. CONFIG.SYS after THINLAPS Command

In case of more complex configurations of the LAN transport system, the `P:` parameter can be used. This parameter allows the administrator to supply a preconfigured `PROTOCOL.INI` file that will be copied to the target. If the `/P:`

parameter is specified, `THINLAPS` will not use the default `PROTOCOL.INI` file, but the supplied one.

4.11.1.1 Support for Additional Drivers

Additional device drivers are shipped with MPTS on the Additional Network Adapter Support diskette. Additional drivers are also provided with new adapters. These additional device drivers will not be stored on the code server when loading the diskette image as described in “The LAPSDISK Command” on page 164. You can add these drivers manually to the image directory.

4.12 NetView DM/2: The NVDMBDSK Utility

This utility installs a NetView DM/2 CC client code on bootable diskettes. The prerequisite for this utility to function properly is to have MPTS installed (`THINLAPS`). The NetView DM/2 Agent needs to be installed on the OS/2 Warp 4 Diskette 2. To start the installation execute the following command from an OS/2 window.

NVDMBDSK Syntax

```
NVDMBDSK
```

The NetView DM/2 Boot Diskette Update window will be displayed as shown in Figure 37 on page 119.

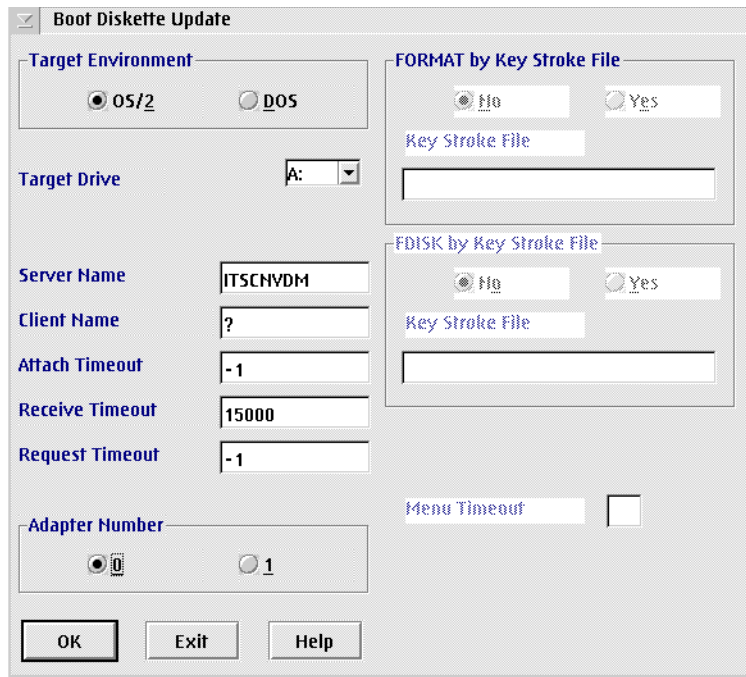


Figure 37. NVDMBDSK - Boot Diskette Update Window

NVDMBDSK updates CONFIG.SYS, which usually resides on the last OS/2 boot diskette. Since OS/2 Warp 4 comes with three boot diskettes in total, the CONFIG.SYS file is located on the boot Diskette 1 (the second one out of the set of three). Depending on the version of NetView DM/2, you may need to copy CONFIG.SYS from Diskette 1 to Diskette 2, run NVDMBDSK, and then copy CONFIG.SYS from Diskette 2 back to Diskette 1.

Updated NVDMBDSK Files

Find an updated version of the NVDMBDSK utility on the enclosed CD-ROM. This NVDMBDSK utility has been changed to prompt the user with a pop-up message requesting to insert the boot diskette with the CONFIG.SYS file before it continues the process.

4.13 General CID Return Codes

Every CID-enabled product should generate the following return codes to the calling program:

- 00 00 Successful program termination occurred.
- 00 04 Successful program processing; warning messages logged.
- 00 08 Successful program processing; error messages logged.
- 00 12 Successful program processing; severe error messages logged.
- 08 00 Data resource was not found.
- 08 04 Data resource access denied because resource is already in use.
- 08 08 Data resource access denied because authorization is missing.
- 08 12 Data path was not found.
- 08 16 Product is not configured.
- 12 00 Storage medium exception (I/O error) occurred.
- 12 04 Device is not ready.
- 12 08 Not enough disk space is available.
- 16 00 Incorrect program invocation was used.
- 16 04 Unexpected condition occurred.

If you have problems, refer to SetCIDType for information on how to make LAN CID Utility treat return codes 00 04, 00 08, and 00 12 as bad return codes.

- FE 00 Successful program processing; reboot queued but no callback required.

FE 01 - FE 03, FE 05 - FE 07, FE 09 - FE 011, FE 13 - FE 3F have generally the same meaning, but special meanings can be assigned by the product.
- FE 04 Successful program processing; warning messages logged; reboot queued but no callback required.
- FE 08 Successful program processing; error messages logged; reboot queued but no callback required.
- FE 12 Successful program processing; severe error messages logged; reboot queued but no callback required.

FE 40 - FE 7F

Successful Program Termination - No Reboot required; special meanings can be documented.

FE 80 - FE FF

Unsuccessful Program Termination - No Reboot required; special meanings can be documented.

FF xx Successful program processing; reboot queued and callback required.

LAN CID Utility manages the state of the product installation by validating the state that the product installation program returns.

LAN CID Utility takes the following steps:

When an exiting product installation program requests a reboot with callback, that program must set the correct byte of the return code to its next state; xx can range from X'00' to X'FF'. For example, on the initial call from LAN CID Utility, the state variable is X'00' and the product installation program can change this value for each reboot request. This value returns to the installation program when LAN CID Utility calls it back.

LAN CID Utility reboots the workstation, retrieves the product installation state parameter, stores it, and passes it to the invoked product installation program by means of the `REMOTE_INSTALL_STATE` state variable.

FD 00 Reserved RC.

4.14 Desktop Shuffler

When installing OS/2 using remote installation through LCU, NVDM/2 or TME 10 SD 3.1.3, each OS/2 Warp 4 component's installation program is executed separately. Installation programs create their own program folders and program icons and place them anywhere on the desktop. The desktop of a local install of OS/2 Warp 4 looks nice and tidy, whereas the desktop of a remote install looks messy.

A tool called Desktop Shuffler is shipped with OS/2 Warp 4. This tool is in charge of rearranging the desktop by moving, renaming, deleting, and creating icons and folders on the desktop. The Desktop Shuffler is executed at the end of the local installation as illustrated in Figure 16 on page 54.

The Desktop Shuffler executable file name is CLNDESK.EXE in the \IBMINST directory. The \IBMINST directory is created automatically under the local install, but when executing a CID-based install, the tool has to be copied to the CID structure.

CLNDESK.EXE Syntax

```
CLNDESK <control_file> <keywords_file>
```

For example:

```
CLNDESK SHUF_CTL.INI SHUF_KEY.INI
```

The control file contains the names of the objects to create, move, shadow or delete. The keywords file contains the keywords to control the Desktop Icon Shuffter execution.

Note

CLNDESK requires NPRESDLL.DLL file to run. NPRESDLL.DLL must be on the LIBPATH or in the working directory.

4.14.1 Example of the SHUF_CTL.INI Control File

The sample file used here demonstrates the format of the Desktop Shuffler control file. The actual file used to arrange the desktop during the local installation is represented in Chapter 4.14.2, “The Working Control and Keywords Files” on page 125. The control file is a plain ASCII file, where all the lines starting with semicolon (;) are comment lines. The file is structured in sections similar to common INI files. The sections are executed in the following order:

- System
- Create
- Delete
- Update
- Move
- Shadow
- Copy

The system section can either be empty or include commented lines. The create section syntax is *Object Class*, <Object ID>, "Title", <Action>, <Location>, where <Object Class> and <Object ID> must be a valid WPS class

name and object name. Title can be any string; action code can be FAIL, REPLACE or UPDATE. For example

```
; This is a sample control file for Desktop Icon Shuffler
;
; Lines starting with a semicolon are comment lines
;
[system]
; This section can be either empty or all comment lines
;
[create]
; Object creation
; Object Class, <Object ID>, "Title", <ACTION>
;
; The following command replaces the existing folder
WPFOLDER, <Sample_folder>, "Sample folder", REPLACE, <WP_DESKTOP>
;
; The following row will create a folder named "Sample folder" and will fail
; if the object with this name already exists
WPFOLDER, <Sample_folder>, "Sample folder", FAIL, <WP_DESKTOP>
;
; The following command updates the existing object or creates a new object
; if the object does not exist
WPFOLDER, <Sample_folder>, "Sample folder", UPDATE, <WP_DESKTOP>
;
; The following command will create a new program item to the Install/Remove folder
WPPROGRAM, <My_program>, "My program", REPLACE, <Sample_folder>,
    EXENAME=C:\EXE\MYPROG.EXE;ICONFILE=C:\EXE\MYPROG.ICO
```

The delete section structure is very simple. The only thing that is required is the object name to be deleted. Here is a sample section:

```
[delete]
; Delete the folder that was added in the create section
<Sample_folder>
```

The update section is used to update object properties. The syntax is one of the following

- <ObjectID> keyword=value
- FOLDER <ObjectID> keyword=value
- SETUPSTRING <ObjectID> keyword=value

Multiple keywords can be defined by separating those by semicolon (;). The sample update section can appear as follows:

```
[update]
; The following line sets the Sample folder default view to tree view
FOLDER <Sample_folder> DEFAULTVIEW = TREE
;
; The following line changes the program title for Sample program
<Sample_program> Title = New program
```

The move section syntax is <ObjectID>, <Destination ObjectID>. The object is not moved, if it already exist in the destination object. Here is an example:

```
[move]
; The following lines move object <My_program> to the desktop
```

```
<MYProgram>, <WP_DESKTOP>
```

The shadow section contains two parameters: *<ObjectID>*, *<Destination ObjectID>*. For example:

```
[shadow]
; The following line creates a shadow on the desktop
<Sample_program>, <WP_DESKTOP>
```

The copy section syntax is *<ObjectID>*, *<Destination ObjectID>*. The sample copy section could look like this:

```
[copy]
;The following line copies the program object to the Install/Remove folder
<Sample_program>, <WP_INSTREMFOLDER>
```

Supported keywords for non-folder objects to be used in create and update sections are listed here:

- CCVIEW=DEFAULT | YES | NO
- DEFAULTVIEW = SETTINGS | DEFAULT | 0-9
- ICONFILE = iconfile.ICO
- ICONPOS = x,y //values between 0 and 100
- MINWIN = HIDE | VIEWER | DESKTOP
- NOCOPY = YES | NO
- NODELETE = YES | NO
- NODRAG = YES | NO
- NODROP = YES | NO
- NOLINK = YES | NO
- NOMOVE = YES | NO
- NOPRINT = YES | NO
- NORENAME = YES | NO
- NOSETTINGS = YES | NO
- NOSHADOW = YES | NO
- NOTVISIBLE = YES | NO
- OPEN = SETTINGS DEFAULT
- TEMPLATE = YES | NO
- TITLE = Title_string

Note

The Desktop Shuffler ignores most lines that are grammatically incorrect. For example, the following action would be ignored because of missing commas:

```
[create]
WPFOLDER, <Folder1>, "New Folder" UPDATE <WP_DESKTOP>
```

4.14.2 The Working Control and Keywords Files

The working Desktop Shuffler control file and keywords file are included on the media distributed with this redbook. The files are located in the \RSP\SHUFFLER directory. These files contain the operations to update the desktop after the installation process to make it look like the desktop created using the local install.

The keywords file is used to control, which #if...#endif sections are executed. You can edit the `Products=` line in the keywords file to exclude or include components. If you, for example, do not want to create MPTS icons, remove the keyword MPTS from the `Products=` line. Here is the SHUF_KEY.INI keywords file included on the CD-ROM:

```
;
; The Keyword file the for ClnDesk.EXE to control execution
; Valid keywords: OS2PEER TCPIP MPTS LDR NW NETFIN SVAGENT MFS
;
Products=OS2PEER TCPIP MPTS LDR NW NETFIN SVAGENT MFS
;
;
```

Hint

The `Products=keyword` list can be a null string.

Several icons are created, moved, shadowed, and renamed when executing the Desktop Shuffler control file. You can add your own definitions to customize the desktop to your own needs. The Desktop Shuffler can be run multiple times using different control and keywords files.

Note

The control file included here requires, that the \IBMINST directory exists on the Warp installation partition. Some programs, like Selective Install for Networking, are run from IBMINST directory, and some program icons will not work.

The same restriction applies to other components. For example, if you did not install the LDR client and the LDR keyword is present in the keywords file, the LDR icons are created on the desktop.

Here is the listing of the actual SHUF_CLT.INI control file used for arranging the desktop after the installation:

```
;
;
; This control file sets the Desktop objects for a CID-installed
; OS/2 Warp 4 system to the same
; arrangements as an attended installed system
```

```

;
;
[system]
;
;
[create]
;
;
; Networking product install icons
; "Selective Install^for Networking"
WPPProgram, <CLTSTART>, "Selective Install^for Networking", REPLACE,
<WP_INSTREMFOLDER>, EXENAME=\IBMINST\NPCONFIG.EXE;ICONFILE=\IBMINST\ICONS\1091.ICO
; "Remove Installation^for Networking"
WPPProgram, <CLTRMOVE>, "Remove Installation^for Networking",
REPLACE, <WP_INSTREMFOLDER>,
EXENAME=\IBMINST\NPRMINST.EXE;ICONFILE=\IBMINST\ICONS\1093.ICO
; "OS/2 Warp^Remote Install"
WPPProgram, <CLT_REMOTE>, "OS/2 Warp^Remote Install", REPLACE,
<WP_INSTREMFOLDER>, EXENAME=\IBMINST\NPSETUP.EXE;ICONFILE=\IBMINST\ICONS\1090.ICO
;
;
;
;
; #if MPTS
; Create MPTS^Network Adapters^and Protocol Services
WPPProgram, <NTS_MPTS_ICON>, "MPTS^Network Adapters^and Protocol Services", UPDATE,
<WP_CONFIG>, EXENAME=MPTS.EXE
; Create DHCP Monitor
WPPProgram, <WS_DHCP_MONITOR>, "DHCP Monitor", UPDATE, <WP_CONFIG>
; Create DDNS Configuration
WPPProgram, <MPTS_DDNS_OBJECT_SETUP>, "DDNS Configuration", UPDATE, <WP_CONFIG>
; Create MPTS Guide
WPPProgram, <MPTSCfg>, "MPTS Guide", REPLACE, <WP_TASKSINFO>
; Create TCP/IP DHCP^Client Administration Guide
WPPProgram, <NP_DHCP_GUIDE>, "TCP/IP DHCP^Client Administration Guide", REPLACE,
<WP_TASKSINFO>
; Create TCP/IP Dynamic IP Introduction
WPPProgram, <NP_DIP_INTRO>, "TCP/IP Dynamic IP Introduction", REPLACE, <WP_TASKSINFO>
#endif
;
; Remote access client folder
; #if LDR
; Create network folder, if it does not exist
WPFolder, <WP_NETWORK>, "Remote Access Client", UPDATE, <WP_CONNECTIONSFOLDER>
; Create a Network Services folder, if it does not exist
WPFolder, <WC_NETSERV>, "Network Services" , UPDATE, <WP_NETWORK>,
SHOWALLINTREEVIEW=YES
; Now the Remote Access icon is being moved directly to Network Services
; Create a Remote access Client folder in Network Services
; WPFolder, <WC_REMOTEACCESS>, "Remote Access Client", UPDATE, <WC_NETSERV>
; Create Remote Access Client Guide
WPPProgram, <LD_CLIENT_GUIDE>, "Remote Access Client Guide", UPDATE, <WP_TASKSINFO>
; Create LAN Distance Advanced Guide
WPPProgram, <LD_ADV_GUIDE>, "LAN Distance Advanced Guide", UPDATE, <WP_TASKSINFO>
; Create LAN Distance remove object
WPPProgram, <LDCS_REMOVE>, "LAN Distance Remove", UPDATE, <WP_INSTREMFOLDER>,
EXENAME=LDREMOVE.EXE;ICONFILE=\IBMINST\ICONS\1033.ICO
; Create shadow of LAN Distance icon in system setup
WPSHADOW, <WP_WALX_SHAD>, "LAN Distance", UPDATE, <WP_CONFIG>, SHADOWID=<WP_WALX>
#endif
;
;
;

```

```

;
#if NW
; Create a Network Services folder, if it does not exist
WPFolder, <WC_NETSERV>, "Network Services", UPDATE, <WP_NETWORK>, SHOWALLINTREEVIEW=YES
; Create program - Delete NetWare Requestor
WPProgram, <CB_NetWare_REMOVE>, "Delete NetWare Requestor", UPDATE, <WP_INSTREMFOLDER>
; Create program - Delete Network Signon coordinator
WPProgram, <NSC_REMOVE>, "Delete Network Signon Coordinator", UPDATE,
<WP_INSTREMFOLDER>, EXENAME=\IBMINST\DELNSC.CMD
#endif
;
#if OS2PEER
; Create a Network Services folder, if it does not exist
WPFolder, <WC_NETSERV>, "Network Services", UPDATE, <WP_NETWORK>, SHOWALLINTREEVIEW=YES
; shadow of readme
WPSHADOW, <LS_README_PEER_SHAD>, "Peer Services Readme", UPDATE,
<WP_READMEFOLDER>, SHADOWID=<LS_README_PEER>
; Create program - Delete Network Signon coordinator
WPProgram, <NSC_REMOVE>, "Delete Network Signon Coordinator", UPDATE,
<WP_INSTREMFOLDER>, EXENAME=\IBMINST\DELNSC.CMD
#endif
;
#if NETFIN
; NetFinity client guide
WPProgram, <NFCLIENTGUIDE>, "NetFinity Client Guide", UPDATE, <WP_TASKSINFO>,
EXENAME=VIEW.EXE;PARAMETERS=\OS2\BOOK\SERVICES.INF
; shadow of readme
WPSHADOW, <NFREADME_SHAD>, "NetFinity Client Readme", UPDATE,
<WP_READMEFOLDER>, SHADOWID=<NFREADME>
#endif
;
[delete]
;
#if SVAGENT
; delete DMI and DPI Programmers guides
<SVADPI>
<SVAPRG>
#endif
;
[move]
; VoiceType will be done here for Beta only:
; Move VoiceType folder to Programs
<WP_SPEECH>, <WP_PROGRAMSFOLDER>
; Move VoiceType Users Guide to Tasks
<SPCH_USERGUIDE>, <WP_TASKSINFO>
;
; Security Services into System Setup
<UPM_FOLDER>, <WP_CONFIG>
;
#if MPTS
; MPTS icon into system setup (in case Peer created it)
<NTS_MPTS_ICON>, <WP_CONFIG>
#endif
;
#if OS2PEER
; Move LS install/config to Install/Remove folder
<LS_INSTALL>, <WP_INSTREMFOLDER>
; Move Peer user guide to Tasks
<PEERUser>, <WP_TASKSINFO>
; Move Main Peer folder to Network Services
<LS_FOLDER>, <WC_NETSERV>
; Move workstation to Network Services
<PEER_WKST>, <WC_NETSERV>

```

```

; Move LS Admin GUI to Network Services
<LS_ADMIN>, <WC_NETSERV>
; Move File and Print Audit log to Utilities
<LS_AUDIT>, <WP_TOOLS>
; Move File and Print Error log to Utilities
<LS_ERROR>, <WP_TOOLS>
; Move Network DDE & Clipboard to Utilities
<LS_CLIP>, <WP_TOOLS>
; Move Network Messaging to Utilities
<LS_NETMSG>, <WP_TOOLS>
; Move Password Coordination Guide to Tasks
<NSC_REF>, <WP_TASKSINFO>
; Move Password Coordination Folder to Utilities
<NSC_FOLDER>, <WP_TOOLS>
#endif
;
; #if NW
; Move NetWare install to Install/Remove
<NVL_INSTALL>, <WP_INSTREMFOLDER>
; Move NetWare Client Guide to Tasks
<NVL_CLIENT>, <WP_TASKSINFO>
; Move Main Novell folder to Network Services
<NOVELL_FOLDER>, <WC_NETSERV>
; Move Password Coordination Guide to Tasks
<NSC_REF>, <WP_TASKSINFO>
; Move Password Coordination Folder to Utilities
<NSC_FOLDER>, <WP_TOOLS>
#endif
;
; #if LDR
; Move Remote Access program into Remote Access Folder
<WP_WALX>, <WC_NETSERV>
#endif
;
; #if SVAGENT
; Move systemview startup out of startup folder, into main SysView folder
<CASOS2>, <KARATCA>
; Move SystemView Agent Remove to Install/Remove
<CAGINSTS>, <WP_INSTREMFOLDER>
; Move SystemView Management Agent Guide to Tasks
<SVAUSR>, <WP_TASKSINFO>
; Move SystemView Agent System Management Agent to Utilities
<KARATCA>, <WP_TOOLS>
#endif
;
; #if NETFIN
; Move PC SystemView System Management Client to Utilities
<RPSM_FLD>, <WP_TOOLS>
#endif
;
; #if MFS
; Move MFS Status: to Mobile Office Services folder
<MFSOBJECT>, <MARSCM1>
; Move Mobile office Services Guide to Tasks
<MCMDOC>, <WP_TASKSINFO>
; Move Mobile Office Services folder to Network Services
<MARSCM1>, <WC_NETSERV>
#endif
;
;
[shadow]

```



```

;
;
; [copy]
;
;
; [update]
;
;
;
; #if MPTS
;   rename MPTS to MPTS^Network Adapters^and Protocol Services
;   <NTS_MPTS_ICON> TITLE=MPTS^Network Adapters^and Protocol Services
;   assign exe file for DHCP Monitor
;   SETUPSTRING <WS_DHCP_MONITOR> EXENAME=DHCPMON.EXE
;   assign exe file for DDNS Configuration
;   SETUPSTRING <MPTS_DDNS_OBJECT_SETUP> EXENAME=DDNSCFG.EXE
;   Set exe and inf file name for NAPS Guide
;   SETUPSTRING <MPTSCfg> EXENAME=VIEW.EXE
;   SETUPSTRING <MPTSCfg> PARAMETERS=\OS2\BOOK\MPTSCFG.INF
;   Set exe and inf file name for DHCP Admin guide
;   SETUPSTRING <NP_DHCP_GUIDE> EXENAME=VIEW.EXE
;   SETUPSTRING <NP_DHCP_GUIDE> PARAMETERS=\OS2\BOOK\DHCPCLNT.INF
;   Set exe and inf file name for Dynamic IP Intro
;   SETUPSTRING <NP_DIP_INTRO> EXENAME=VIEW.EXE
;   SETUPSTRING <NP_DIP_INTRO> PARAMETERS=\OS2\BOOK\DIPINTR.INF
; #endif
;
; #if NETFIN
;   Rename NetFinity to NetFinity System Management Client
;   <RPSM_FLD> TITLE=NetFinity^System Management Client
;   TME 10 NetFinity^System Management Client^Readme
;   <NFPREADME_SHAD> TITLE=TME 10 NetFinity^System Management Client^Readme
; #endif
;
; #if SVAGENT
;   Rename SystemView folder
;   <KARATCA> TITLE=SystemView Agent^System Management Agent
;   Rename Deinstall utility to SystemView Agent Remove
;   <CAGINSTS> TITLE=SystemView Agent Remove
;   Rename SystemView Management Agent Guide
;   <SVAUSR> TITLE=SystemView Management Agent Guide
; #endif
;
; #if NW
;   Rename Netware Services folder
;   <NOVELL_FOLDER> TITLE=Netware Services
;   Assign cmd file for NetWare Remove
;   SETUPSTRING <CB_NetWare_REMOVE> EXENAME=\IBMINST\DELNW.CMD
;   Rename NetWare Client Install
;   <NVL_INSTALL> TITLE=NetWare Client Install
;   Rename NSC folder Password^Coordination
;   <NSC_FOLDER> TITLE=Password^Coordination
;   Rename SignOn Password^Coordination
;   <NSC_PM> TITLE=Password^Coordination
;   Rename Server Password Coordination^Server
;   <NSC_SVR> TITLE=Password Coordination^Server
;   Rename Retry Password Coordination^Retry Process
;   <NSC_RETRY> TITLE=Password Coordination^Retry Process
;   Rename Reference Password Coordination Guide
;   <NSC_REF> TITLE=Password Coordination Guide
;   Assign cmd file for Password Coordination Remove
;   SETUPSTRING <NSC_REMOVE> EXENAME=\IBMINST\DELNSC.CMD
; #endif

```

```

;
; #if OS2PEER
; rename LS install/config
;   <LS_INSTALL> TITLE=File and Print Client^Install/Remove
; rename OS/2 Peer to Logons
;   <LS_FOLDER> TITLE=Logons
; rename Start OS/2 Peer to Start^File and Print Client
;   <LS_START> TITLE=Start^File and Print Client
; rename Peer Workstation Logon to File and Print Client^Workstation Logon
;   <PEER_LOCALLOGON> TITLE=Print Client^Workstation Logon
; readme Network^Readme
;   <LS_README_PEER_SHAD> TITLE=Network^Readme
;
; Rename NSC folder Password^Coordination
;   <NSC_FOLDER> TITLE=Password^Coordination
; Rename SignOn Password^Coordination
;   <NSC_PM> TITLE=Password^Coordination
; Rename Server Password Coordination^Server
;   <NSC_SVR> TITLE=Password Coordination^Server
; Rename Retry Password Coordination^Retry Process
;   <NSC_RETRY> TITLE=Password Coordination^Retry Process
; Rename Reference Password Coordination Guide
;   <NSC_REF> TITLE=Password Coordination Guide
; Assign cmd file for Password Coordination Remove
;   SETUPSTRING <NSC_REMOVE> EXENAME=\IBMINST\DELNSC.CMD
; #endif
;
; #if MFS
; rename Mobile Office Services
;   <MARSCM1> TITLE=Mobile Office Services
; rename Start MFS
;   <MCMPPROG> TITLE=Start^Mobile Office Services
; rename Stop MFS
;   <MCMSTOP> TITLE=Stop^Mobile Office Services
; rename MFS doc
;   <MCMDOC> TITLE=Mobile Office Services Guide
; #endif
;
; #if LDR
; Rename Remote Access Client
;   <WP_WALX> TITLE=Remote Access Client
; Set exe and inf file name for Remote Access Client Guide
;   SETUPSTRING <LD_CLIENT_GUIDE> EXENAME=VIEW.EXE
;   SETUPSTRING <LD_CLIENT_GUIDE> PARAMETERS=\OS2\BOOK\A9M04MST.INF
; Set exe and inf file name for Remote Access Advanced Guide
;   SETUPSTRING <LD_ADV_GUIDE> EXENAME=VIEW.EXE
;   SETUPSTRING <LD_ADV_GUIDE> PARAMETERS=\OS2\BOOK\A3T12MST.INF
; #endif
;
;
;

```

Find more information about the Desktop Shuffler in Appendix A.5, "The Desktop Shuffler Tool" on page 463.

4.15 Remote Multiple Printer Support

This section describes the use of the remote multiple printer installation application (RMPI). The first part will give a brief description of the program and the second part will describe the command line parameters and the keywords in detail.

4.15.1 Remote Printer Install (RINSTPRN)

The main purpose of the remote multiple printer installation program (RINSTPRN) for OS/2 is installation of the printer-related issues at the time of initial OS/2 installation. The RINSTPRN utility shipped with this book is rewritten from previous OS/2 versions to run on OS/2 Warp 4.

The RINSTPRN utility makes it possible to install multiple printers and queues through a response file instead of going through many dialog panels. It performs the installation of printers, queues and ports and configures communication ports. This application also provides the administrator with the ability to make final adjustments on the client's workstations, including printer-driver-specific information such as job and printer properties, fonts, options, and so forth during the automated process. It also allows the definition of network queues and the definition of WIN-OS/2 printers.

The utility executes either from the `UserExit` or `ExtendedInstall` statements of the OS/2 response file, locally from the command line, from an entry to the LCU command file, or out of a change file in NetView DM/2 or TME 10 SD 3.1.3 for OS/2.

The program reads the response file, interprets it and looks for consistency between the defined queues, printers and other values. After finishing this step, it installs the printers, drivers and queues. All actions are logged into a log file for administrative purposes.

This program makes it possible to administer complex printer and queue configurations without the administrator being at the location for installation.

4.15.2 The RINSPRN Command

`RINSTPRN` executed from the OS/2 command line can be used in conjunction with the following parameters:

RINSTPRN Command Syntax

```
RINSTPRN  /DSC
          /DRV
          /L1
          /R
          /S
          /T
          /WPR
          /WDR
          /WT
```

Each keyword is optional. Keywords can be used in any order. If the same keyword appears twice on the command line, the value of the last occurrence is used. The keywords are followed immediately by a colon (:) and a value. Blanks are not allowed between keyword, colon and value. Keywords are separated by one or more blanks. Misspelled keywords are logged into the log file and simply ignored (the processing will continue). If one of the keywords is not defined in the command line, a default value will be used.

/DSC: This keyword defines the name of the printer description list. A partially or fully qualified OS/2 path name including a drive letter can be used.

Note: If the drive letters A: or B: are used, make sure the driver Diskette 1 is in the specified drive before starting the program.

The PRDESC.LST file changes with every release. A proper printer install can only take place if the PRDESC.LST matches the driver install diskettes. The current version of PRDESC.LST resides on Diskette 1 of the driver diskettes.

Default: PRDESC.LST in the working directory.

For example:

```
RINSTPRN /DSC:X:\IMG\OS2V21\PMDD_1\PRDESC.LST
```

/DRV: This keyword defines the name of the printer driver list. A partially or fully qualified OS/2 path name, including a drive letter, can be used.

Note: If the drive letters A: or B: are used, make sure the driver Diskette 1 is in the specified drive before starting the program.

The PRDRV.LST changes with every release. A proper printer install can only take place if the PRDRV.LST matches the driver install diskettes. The current version of PRDRV.LST resides on Diskette 1 (PMDD_1) of the printer driver diskettes.

Default: PRDRV.LST in the working directory.

For example:

```
RINSTPRN /DRV:X:\IMG\OS2V21\PMDD_1\PRDRV.LST
```

/L1: This keyword defines the location of the log file into which the RINSTPRN program logs its response file analysis, activities and execution results. A partially or fully qualified OS/2 path name, including a drive letter, can be used.

Default: RINSTPRN.LOG in the working directory.

For example:

```
\RINSTPRN /L1:C:\RINSTPRN.LOG
```

/R: This keyword defines the location of the printer install response file. A partially or fully qualified valid OS/2 path name, including a drive letter, can be used.

Note: If the drive letters A: or B: are used, make sure the proper diskette is in the specified drive before starting the program. Please be aware of the fact that when the keywords /DSC and /DRV point to the same drive, all three files have to be on this same diskette.

Default: PRINTER.RSP in the working directory.

For example:

```
RINSTPRN /R:X:\RSP\OS2V21\PRINTER.RSP
```

/S: This keyword defines the source drive and directory where the drivers and fonts to be installed are located. A fully qualified path name with a drive letter can be used. If the drive is A or B, the program asks for the printer driver diskettes on A: or B:. On any other drive (C to Z), the program looks for subdirectories called PMDD_1 to PMDD_n (depending on how many disks are

mentioned in column two of the PRDRV.LST) in the specified directory. This drive can also be a redirected drive.

Default: A:.

For example:

```
RINSTPRN /S:A:
```

/T: This keyword defines the target drive where the OS/2 system is installed. Either just the drive letter or the drive letter with a colon can be specified. Use this keyword if OS/2 has been installed to a logical partition, rather than to a primary partition.

Default: C.

For example:

```
RINSTPRN /T:D
```

/WPR: This keyword defines the name of the WIN-OS/2 printer setup file. A partially or fully qualified OS/2 path name, including a drive letter, can be used.

Note: If the drive letters A: or B: are used, make sure a diskette containing the specified file is inserted in the drive before starting the program.

The default value for this parameter is CONTROL.INF. This file resides in the \OS2\MDOS\WINOS2\SYSTEM subdirectory after an installation of OS/2 and may change with every release. This parameter is only used if an installation of WIN-OS/2 printers is requested in the response file.

Default: CONTROL.INF in the working directory.

For example:

```
RINSTPRN /WPR:X:\EXE\CONTROL.INF
```

/WDR: This keyword defines the name of the map file between OS/2 and WIN-OS/2 device drivers. A partially or fully qualified OS/2 path name, including a drive letter, can be used.

Note: If the drive letters A: or B: are used, make sure a diskette

containing the specified file is inserted in the drive before starting the program.

The default value for this parameter is DRVMAP.INF. This file resides in the \OS2\MDOS\WINOS2\SYSTEM subdirectory after an installation of OS/2 and may change with every release. This parameter is only used if the WIN-OS/2 printer installation to an OS/2 printer is requested in the response file.

Default: DRVMAP.INF in the working directory.

For example:

```
RINSTPRN /WDR:X:\EXE\DRVMAP.INF
```

/WT: This keyword defines the target drive where WIN-OS/2 is installed. Either just the drive letter or the drive letter with a colon can be specified. Use this keyword if WIN-OS/2 has been installed to a logical partition rather than to a primary partition.

Default: C.

For example:

```
RINSTPRN /WT:D
```

The following complete example looks for a printer response file on redirected drive Z: with the name PRINTER.RSP. The PRDRV.LST is located on redirected drive Z: in the root subdirectory \PMDD_1. The PRDESC.LST is located on redirected drive Z: in the root subdirectory \PMDD_1. The WIN-OS/2 printer setup file is located in the Z: directory and has the name CONTROL.INF. The WIN-OS/2 driver map file is located in the Z: directory and has the name DRVMAP.INF. The USERnnnn.LOG file will be written to the redirected drive Z: thereby gathering the install information on the server. OS/2 and WIN-OS/2 are installed on drive D.

```
RINSTPRN /R:Z:\PRINTER.RSP
/DRV:Z:\PMDD_1\PRDRV.LST
/DSC:Z:\PMDD_1\PRDESC.LST
/L1:Z:\USERnnnn.LOG
/S:Z:
/T:D
/WPR:Z:\CONTROL.INF
/WDR:Z:\DRVMAP.INF
/WT:D
```

4.15.3 Remote Printer Install Response File Keywords

All the subsequent keywords begin definitions which include one or more sub-keywords. Each definition ends with an ENDxxx keyword.

ADDPORTR Defines a new port configuration on the system. The new port appears in the list of available ports on the configured system.

Example:

```
ADDPORTR
  NAME=NETPORT1
  PORTDRIVER=NETDRV
  DESCRIPTION=This is a test port
  NETADDR=123.123.123.123
  CFGFILE=C:\os2\install\test.cfg
  CFGNAME=NetPortX
ENDPORTR
```

ADDPORTR sub-keywords:

```
NAME
DESCRIPTION
PORTDRIVER
NETADDR
NETADAPTER
HOSTNAME
ATTACHINTERFACE
CFGFILE
CFGNAME
PNLFILE
REFERENCE.
```

Configuration files (`CFGFILE`) will be processed first so any information can be overwritten by the keywords defined.

If the configuration name is specified (`CFGNAME=PortName`), then `RINSTPRN` will look in the configuration file for the settings for the given port name (in this example `NetPortX`).

If a configuration file is given but no `CFGNAME` is given, then `RINSTPRN` will look in the configuration file for the settings for the new port name (in this

example NETPORT1), and if not found, it will use the first application name in the configuration file.

Note: The name FILE is reserved and will cause an error in ADDPORT

ADDPORTDRIVER Defines the name of the port driver and the path to it.

Example:

```
ADDPORTDRIVER
  NAME=PARALLEL
  SRCPATH=X:\DRIVERS
```

```
ENDPORTDRIVER
```

ADDPORTDRIVER sub-keywords:

NAME

SRCPATH

ADDPRINTDRIVER Specifies a driver to be loaded.

Example:

```
ADDPRINTDRIVER
  NAME=LASERJET.DRV
ENDPRINTDRIVER
```

ADDPRINTDRIVER sub-keywords:

NAME (truncated to eight characters if longer)

SRCPATH If SRCPATH is defined, it will try to find the driver in .\, .\OS2DRV or .\PMDD-x. If not defined, the command line argument will be used. If none, default will be A:

ADDPRINTDEVICE Defines the print device as defined in the PRDESC.LST file

Example:

```
ADDPRINTDEVICE
  NAME=IBM 4029 Laserprinter 10
  DRIVER=LASERJET
ENDPRINTDEVICE
```

ADDPRINTDEVICE sub-keywords:

NAME

DRIVER Must be the NAME defined in ADDPRINTDRIVER

ADDDEVICE Defines a new device

Example:

```
ADDDEVICE
  NAME=IBM LASERJET
  LOGADDRESS=LPT1
  DRIVERS=LASERJET.IBM 4029 Laserprinter 10
ENDDEVICE
```

ADDDEVICE sub-keywords:

NAME	(required)
LOGADDRESS	If defined, can be LPT1 to LPT32, COM1 to COM4 or a port defined in <code>ADDPOR</code>
DRIVERS	Must be "driver.device". If not installed a new driver and device will be created as with <code>ADDPRINTDRIVER</code> and <code>ADDPRINTDEVICE</code>).
COMMENT	
DESCRIPTION	
REFERENCE	Must be an already-defined device.

ADDQUEUE Defines a new queue

Example:

```
ADDQUEUE
  NAME=TEST1
  DRIVERNAME=LASERJET.IBM 4029 Laserprinter 10
  PRINTERS=IBM LASERJET
ENDQUEUE
```

ADDQUEUE sub-keywords:

NAME	Physix=cal name of the queue (required}. Truncated to eight characters if necessary.
DESCRIPTION	Will be the title of the print object. Should be coded; otherwise no object's title will appear.
COMMENT	Will be the title of the print object. Should be coded; otherwise no object's title will appear.
SEPPFILE	gives the name of a file that will be used to define a separator file.

PRIORITY	Must be between 1 and 9.
STARTTIME	In minutes after midnight.
STOPTIME	In minutes after midnight.
HOLD	Y or N (default: N).
APPDEFAULT	Y or N (default: N).
PRINTERSPECIFIC	Y or N (default: N).
PRINTWHILESPool	Y or N (default: Y).
PRINTPROCESSOR	Default: PMPRINT.
DRIVERNAME	Must be "driver.device". If not installed, a new driver and device will be created as with <code>ADDPRINTDRIVER</code> and <code>ADDPRINTDEVICE</code>).
PRINTERS	Must be an installed device.
PORT	If defined can be LPT1 to LPT32, COM1 to COM4 or a port defined by <code>ADDPOR</code> .
REFERENCE	Points to a previously defined queue.
DEFINEQUEUEPROPS	Defines the queue properties .
ENDQUEUEPROPS	Ends the queue properties definition.
DEFINEQUEUEPROPS sub-sub-keywords:	
ORIENTATION	Valid values are: Portrait, Landscape, Rev_Portrait, Rev_Landscape.
COLOR	Valid values are: Monochrome, Color.
PRINTQUALITY	Valid values are: Draft, Low, Medium, High.
DUPLEX	Valid values are: Off, Book, Flip.
COLLATE	Valid values are: Off, On.
FEED	Valid values are: Manual, Automatic.
FORMFEEDCONTROL	Valid values are; None, Conditional, Compulsory.
COPIES	Number of copies to print.
SCALING	
NUP	Number of pages per sheet
RESOLUTION	Values are X-res[,Y-res].
	If only one value, Y-res is equal to X-res.

FORMNAME	Valid values are: None, Letter, Legal, Wide, Csheet, Dsheet, Esheet, LetterSmall, Tabloid, Ledger, Statement, Executive, A0, A1, A2, A3, A4, A4_Small, A5, B4, B5, Folio, Quatro, 10x14, 11x17, Note, Env_9, Env_10, Env_11, Env_12, Env_14, Env_dl, Env_a2, Env_c3, Env_c4, Env_c5, Env_c6, Env_c65, Env_c9, Env_c10, Env_b4, Env_b5, Env_b6, Env_Italy, Env_Monarch, Env_Personal, Fanfold_us, Fanfold_Std_German, Fanfold_Lgl_German, Architect_Bsheet, Architect_Csheet, Architect_Dsheet, Architect_Esheet, Card_A6, Card_4x6, Card_5x8, Card_Hagaki, Label_Standard, Label_Shipping, Label_Disk, Label_Euro.
TRAYNAME	Valid values are: None, Upper, Lower, Middle, Manual, Envmanual, Auto, Tractor, Smallfmt, Largefmt, LargeCapacity, Cassette.
MEDIANAME	Valid values are: None, Plain, Transparency, Glossy, Special, Coated, Backprint, Cloth, Thick, Stationary, Envelope, Continuous_Long, Continuous_Short, Tab_Stock, Multi_Part_Form, Labels.
ADDNETQUEUE	Creates a network queue.

Example:

```
ADDNETQUEUE
NAME=OPTRA
COMMENT=Printer on NET
REMOTECOMPUTER=LS:\\ITSCPS00
REMOTEQUEUE=OPTRA
DRIVERNAME=PSCRIPT.Lexmark Optra N
ENDNETQUEUE
```

ADDNETQUEUE sub-keywords: All the ADDQUEUE keywords plus the following:

REMOTECOMPUTER	Required value must be of the form LS:\\computer.
REMOTEQUEUE	Required value must be the name of the remote queue.
SYNCPRINTERPROPS	Values are Y or N. Default is Y.
SYNCJOBPROPS	Values are Y or N. Default is N.

Note: NAME and DRIVERNAME (defined in ADDQUEUE) are required keywords. A COMMENT or DESCRIPTION keyword should be coded too, to give a name to the created object.

DELETEDPORT Will delete a defined port.

Example:

```
DELETEDPORT
  NAME=MYPORT
ENDPORT
```

DELETEDPORT sub-keyword:

NAME The value is the name of a defined port.

DELETEDDEVICE Will delete a defined device.

Example:

```
DELETEDDEVICE
  NAME=MYDEVICE
ENDDEVICE
```

DELETEDDEVICE sub-keyword:

NAME Value is the name of a defined device.

DELETEQUEUE Will delete a defined queue.

Example:

```
DELETEQUEUE
  NAME=MYQUEUE
ENDQUEUE
```

DELETEQUEUE sub-keyword:

NAME Value is the name of a defined queue.

In the following example of an RINSTPRN response file, previous resources are deleted first. If they don't exist, a warning message appears in the log file. Then definitions are made for three different local printers with their own driver and a fourth remote printer:

```
DELETEQUEUE
  NAME=JETQUEUE
ENDQUEUE
*
DELETEQUEUE
  NAME=OPTRAPCL
```

```

ENDQUEUE
*
DELETEQUEUE
  NAME=OPTRASCR
ENDQUEUE
*
DELETEDEVICE
  NAME=IBM LASERJET
ENDDEVICE
*
DELETEDEVICE
  NAME=OPTRA PCL5
ENDDEVICE
*
DELETEDEVICE
  NAME=OPTRA POSTSCRIPT
ENDDEVICE
*
ADDPRINTDRIVER
  NAME=LASERJET
ENDPRINTDRIVER
*
ADDPRINTDRIVER
  NAME=IBMPCL5
ENDPRINTDRIVER
*
ADDPRINTDRIVER
  NAME=PSCRIPT
ENDPRINTDRIVER
*
ADDPRINTDEVICE
  NAME=IBM 4029 Laserprinter 10
  DRIVER=LASERJET
ENDPRINTDEVICE
*
ADDPRINTDEVICE
  NAME=Lexmark Optra N
  DRIVER=IBMPCL5
ENDPRINTDEVICE
*
ADDPRINTDEVICE
  NAME=Lexmark Optra N
  DRIVER=PSCRIPT
ENDPRINTDEVICE
*
ADDDEVICE
  NAME=IBM LASERJET
  LOGADDRESS=LPT1
  DRIVERS=LASERJET.IBM 4029 Laserprinter 10
ENDDEVICE
*
ADDDEVICE
  NAME=OPTRA PCL5
  LOGADDRESS=LPT2
  DRIVERS=IBMPCL5.Lexmark Optra N
  COMMENT=Optra in PCL5 mode
ENDDEVICE
*
ADDDEVICE
  NAME=OPTRA POSTSCRIPT
  LOGADDRESS=LPT3
  DRIVERS=PSCRIPT.Lexmark Optra N
  DESCRIPTION=Optra in postscript mode

```

```

ENDDEVICE
*
ADDQUEUE
  NAME=JETQUEUE
  DRIVERNAME=LASERJET.IBM 4029 Laserprinter 10
  PRINTERS=IBM LASERJET
ENDQUEUE
*
ADDQUEUE
  NAME=OPTRAPCL
  DRIVERNAME=IBMPCL5.Lexmark Optra N
  PRINTERS=OPTRA PCL5
  COMMENT=QUEUE for OPTRA in PCL5 mode
  HOLD=Y
  PORT=LPT32
  DEFINEQUEUEPROPS
    ORIENTATION=LANDSCAPE
    RESOLUTION=600
    FORMNAME=A4
  ENDQUEUEPROPS
  ADDWINOS2=Y
ENDQUEUE
*
ADDQUEUE
  NAME=OPTRASCR
  DRIVERNAME=PSCRIPT.Lexmark Optra N
  PRINTERS=OPTRA POSTSCRIPT
  DESCRIPTION=QUEUE for OPTRA in Postscript mode
  DEFINEQUEUEPROPS
    PRINTQUALITY=Low
    COPIES=2
    NUP=2
  ENDQUEUEPROPS
  ADDWINOS2=Y
ENDQUEUE
*
ADDNETQUEUE
  NAME=OPTRA
  COMMENT=Printer on NET
  REMOTECOMPUTER=LS:\\ITSCPS00
  REMOTEQUEUE=OPTRA
  DRIVERNAME=PSCRIPT.Lexmark Optra N
  APPDEFAULT=Y
ENDNETQUEUE
*

```

4.15.4 RINSTPRN Response File Configurator RINSTCFG

RINSTCFG.EXE is a utility to help creation of RMPI response file. It is a PM application. This configurator can be found on the CD-ROM included with this redbook.

To run it, just enter RINSTCFG. The first screen is empty; you will have to create the resources you want to get in your response file.

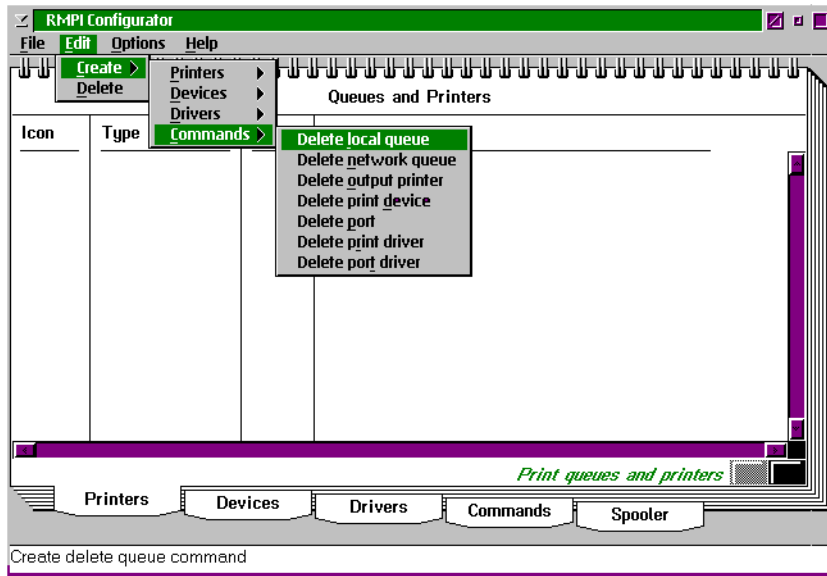


Figure 38. RMPI Configurator Window

In the example shown in Figure 39 on page 144, we clicked on **Edit**, then **Create**, then **Drivers** then **Printer Driver**.

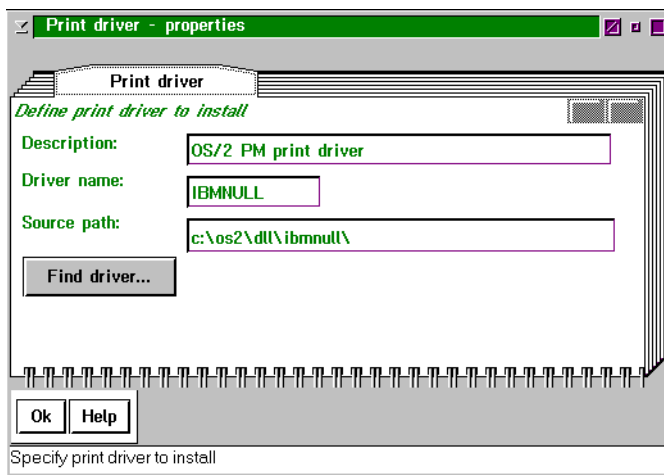


Figure 39. Print Driver - Properties Window

The source path is only useful if you want to find the existing .DRV files, or you can just type in the driver name you want, for example: PSSCRIPT.

In the example shown in Figure 40, we clicked on **Edit**, then **Devices**, then **Printer Driver Device** and double-clicked on the line with IBMNULL driver. We can change the title and choose the right device in the list.

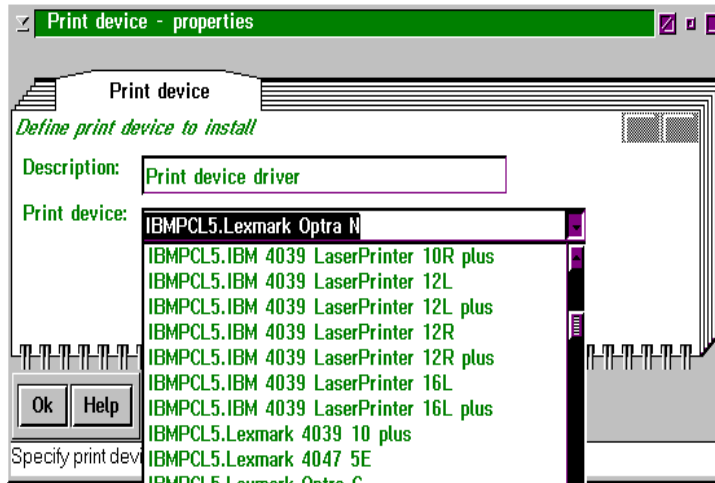


Figure 40. Print Device - Properties Window

In the previous example, IBMPCL5 was not selected as a driver to load. The configurator will add it to the response file.

In example shown in Figure 41, we clicked on **Edit, Create, Printers** and **Local Queue**. Same as before we had to choose the right print driver in the list and change name and description.

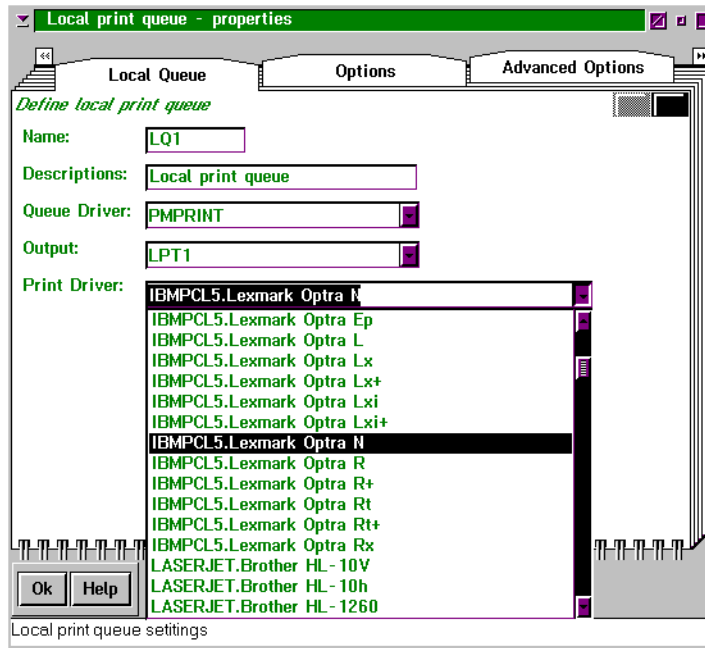


Figure 41. Local Print Queue - Properties Window

Don't forget to click on the **Options** page to select the options, such as **Hold the queue**, that are appropriate for the printer queue.

If you want to delete existing elements, you can choose **Edit, Create, Commands** then **Delete** the right element as shown in Figure 38 on page 144.

Note: Don't forget to **SAVE** before quitting configurator.

Is this example, the created response file is the following:

```

*
* Spooler
*
SPOOLERPROPS
  DESCRIPTION = Spooler
  SYNCHATLOGON = N
ENDSPOOLERPROPS

*
* Port definitions
*
ADDPOR
  NAME = LPT1
ENDPOR

```

```

*
* Print driver definitions
*
ADDPRINTDRIVER
    NAME = PSSCRIPT
    DESCRIPTION = OS/2 PM print driver
ENDPRINTDRIVER

*
* Print device definitions
*
ADDPRINTDEVICE
    NAME = Lexmark Optra N
    DESCRIPTION = Print device driver
    DRIVER = IBMPCL5
ENDPRINTDEVICE

*
* Queue definitions
*
ADDQUEUE
    NAME = LQ1
    COMMENT = Local print queue
    DRIVERNAME = IBMPCL5.Lexmark Optra N
    PRINTPROCESSOR = PMPRINT
    SETUP = FOLDER=<WP_PRINTERSFOLDER>;
    PORT = LPT1
    STARTTIME = 0
    STOPTIME = 0
    HOLD = Y
    DEFINEQUEUEPROPS
        COPIES = 1
        SCALING = 1
        NUP = 1
    ENDQUEUEPROPS
ENDQUEUE

```

4.15.5 Backing Up and Restoring Printer and Job Properties

In order to use printer and job properties during the remote installation, a property file has to be created first. This section describes how to create this property file and how to use it without a remote installation.

4.15.5.1 Backing Up Printer and Job Properties

A printer and job properties file consists of printer-driver-specific data defined for a printer and a queue. The printer part describes hardware-related things like which fonts are installed, which paper is in the trays, or which options are installed on the printer. The job properties consist of information about which paper to select, which resolution and orientation to use, and so on. So printer properties belong to the printer, and job properties belong to a queue. These two types of properties are closely related to each other; so it makes no sense to back one of them up without the other.

Printer and job properties are backed up using the `BACKPRN` program. They are stored in a file, where they can be restored using the `RESTPRN` program (see Restoring Printer and Job Properties).

Invoking `BACKPRN` without any command line parameter will show the syntax of the program, as well as the available printer, queues and the printer driver used by them. To create a property file, the invocation is:

BACKPRN Syntax and Example

```
BACKPRN <printer-name> [<queue-name>] <file-name>
```

`<printer-name>` The name of the printer to copy the printer properties from
`<queue-name>` (optional)

The name of the queue to copy the job properties from (if no queue is specified, the first defined for the printer is used)

`<file-name>` The name of the property file

For Example:

```
BACKPRN PSCRIPT1.PSCRIPT1 pscript.pjp
```

The property file created with the `BACKPRN` contains the printer and job properties, as well as information about the driver used.

4.15.5.2 Restoring Printer and Job Properties

Printer and job properties can be restored using the `RESTPRN` program. An invocation without specifying any parameter shows the command line syntax as well as a list of available printers, queues and their printer drivers. An invocation specifying a property file and a question mark shows the printer name, queue name and the driver to which the properties stored in the file belong. An invocation with only the name of the property file uses the printer name and queue name stored in the file. If the printer and/or queue does not exist, they will be created by the program. To restore a property file, the command line invocation is:

RESTPRN Syntax and Example

```
RESTPRN <file-name> [<printer-name>[.<queue-name>]]
```

`<file-name>` is the name of the property file

<printer-name> (optional)

The name of the printer to copy the printer properties to. If the printer doesn't exist, it will be created.

If no printer is specified, the name stored in the property file is used.

<queue-name> (optional)

The name of the queue to copy the job properties to. If the queue doesn't exist, it will be created.

If no queue is specified, the name stored in the property file is used.

For Example:

```
RESTPRN pscript.pjp PSCRIPT1.PSCRIPT1
```

4.15.6 Holding or Releasing a Queue

Any print queue can be held or released from the command line using the `CHGQUE` program. An invocation without specifying any parameter shows the command line syntax as well as a list of available printers, queues and their printer drivers. An invocation specifying a queue name shows the actual status of the queue (Held or Released). To change the status of a queue, the command line invocation is:

CHGQUE Syntax and Example

```
CHGQUE <queue-name> [/H[OLD]] [/R[LEASE]]
```

<queue-name> The name of the queue for that the status has to be changed or displayed.

/H[OLD] Hold the queue.

/R[LEASE] Release the queue.

For Example:

```
CHGQUE PSCRIPT1 /H
```

4.16 Software Installer

The Software Installer is an installation and customization tool that can be used by developers to offer a simple method to remotely distribute their own applications.

The following operations can be performed:

- Creating easy-to-configure installation menus and CID-enabled installations
- Building distribution diskettes
- Supplying user exits during the installation process to:
 - Set environment variables
 - Modify CONFIG.SYS and STARTUP.CMD
 - Add or delete information from configuration files (*.INI)
 - Work with Workplace Shell (WPS) object classes, like creation, modification or deletion
 - Create, modify and delete objects
 - Execute another program
 - Work with DLLs
- Tailoring installation panels, animated bitmaps, display information windows about the installation process.
- Replacing product files in use
- Customizing text on help panels

From the user point of view, the Software Installer is a utility that comes with the application software for easy installation and maintenance. It can:

- Install the product in attended mode with a graphical user interface
- Install it lightly attended or unattended, controlled by response files
- Select components of a multi-component product

In the same way, it can also be used for maintenance of the installed products:

- Updating an installed product using a CSD (Corrective Service Diskettes) or the next release of the product
- Restoring the previous level of the product
- Deleting a product from the workstation

4.16.1 Parameter Definitions

Software Installer is called using parameters. Though some parameters may be particular to the product to install, the standard parameters are the following:

INSTALL Syntax

```
INSTALL /A:<action>  
      C:<catalog file name>  
      /G:<general include path>  
      /L1:<error log file>  
      /L2:<history log file>  
      /P:<product name>  
      /R:<response file>  
      /S:<source path>  
      /T:<install target directory>  
      /TU:<update target CONFIG.SYS directory>  
      /X
```

/A: Action to be performed (mandatory)

The action values can be **D** (Delete), **I** (Install), **R** (Restore) or **U** (Update)

/C: Fully qualified catalog file name

Specifies the path and name of the file giving the description of the package.

/G: Fully qualified general path

Specifies the directory to be searched for include files, if they do not reside in any of the other directories accessed by the installation program.

/L1: Error log name

/L2: History log name

/P: Product name

Specifies the name of the product to be installed. Make sure that the spelling of the product name is correct. Be aware that the product names are language- and case-dependent.

/R: Fully qualified path and name of response file

- `/S:` Fully qualified path to the product images
- If omitted, the installation program assumes that the files reside in the same directory from which it is running.
- `/T:` Installation target directory
- Specifies the drive and path where the product files are installed.
- `/TU:` Target CONFIG.SYS directory
- Specifies the drive and path of the target CONFIG.SYS to be updated.
- `/X` Unattended execution (mandatory for CID)

4.16.2 Product Database File

The file EPFIS.INI holds all the information about installed products. This file is created during the installation of the first product that uses the Software Installer.

The file is located in the \OS2\SYSTEM subdirectory or in the subdirectory specified by the `EPFINSTDIR` environment variable.

Every time a product that uses the Software Installer, the product database file is maintained. Do not manually edit this file.

4.16.3 Installation Scenario

Now we will have a look at the Software Installer from the user's point of view. In this scenario, we use NetView DM/2 as the software distribution manager.

1. The images of the software to distribute are loaded into to IMG structure of the code server.
2. The catalog entry for the new product is created, and the install action is defined.
3. Create the response files to fit the workstation.
4. Select the catalog entry and define the workstations to be installed. The installation can be done on several workstations at the same time.
5. The workstation agent will receive an installation request for its server. A connection to the server is established, and the installation process will be started in the background without any user action.
6. The distribution agent starts the INSTALL.EXE with the action, log file, history file, response file, product name and so forth parameters, and waits for the program to return.

INSTALL.EXE does the following:

- Parses the command line
- Creates/opens the logfiles and logs its activity
- Determines the source and target directory
- Finds the hard drive with most space available
- Creates a temporary directory on this hard drive
- Copies the runtime components into the temporary directory
- Unpacks the INSTALL.IN_ file
- Invokes the xxxIDLDS.EXE and passes the appropriate parameters
- Waits for the termination of xxxIDLS.EXE

7. xxxIDLS.EXE does the following:

- Parses the arguments and the response file
- Opens the log files and logs its activity
- Reads the catalog file
- Reads the Software Installer product database file (EPFIS.INI) on the workstation and checks the status of the product
- Reads the package file of the product
- Reads variables form the response file
- Transfers files, updates CONFIG.SYS, creates desktop objects and runs user exits
- Updates the product database file
- Terminates and passes return code

8. After INSTALL.EXE gets back control:

- Cleans up, deletes files from temporary directory and removes them
- Terminates and passes return code

9. The distribution agent receives the return code and updates the NetView DM/2 catalog to represent the changes made to the client.

More information about Software Installer can be found in the following redbook titled *OS/2 Installation Techniques: The CID Guide*, SG24-4295.

4.17 ASD - Alternative Software Delivery

Alternative Software Delivery (ASD) is a new method used by IBM to deliver applications and new functions, called features, to customers. The features are made available through a download mechanism from the Internet and intranet called Software Choice (see Figure 42 on page 154).



Figure 42. OS/2 Software Choice Web Site as of 12/97

Features selected can be installed on a local workstation, or can be integrated in an enterprise software distribution process. ASD is a new technique of distributing software, software updates and fixes. It can make use of OS/2 Netscape's and Web Explorer's built-in RSU (Remote Software Update) function. RSU is an automated install process with minimal user interaction.

A new installer is used to install the features: Feature Installer which, in addition, provides functions such as versioning, uninstalling and protecting from downleveling. There is a command line interface to the feature installer called CLIFI.

The Feature Installer (FI) is integrated into OS/2 Warp 4 and is used to install components that can be found on the OS/2 Warp 4 CD-ROM's \OS2IMAGE\FI directory. Users benefit from a consistent front-end to the install process rather than having to deal with the myriad of today's various installation programs.

Feature Install objects are stored in the \OS2\INSTALL\Installed Features directory. Once you click on the **Installed Features** folder and then **Install Object - Inventory**, the InstallObject - Inventory window is displayed as shown in Figure 43 on page 156.

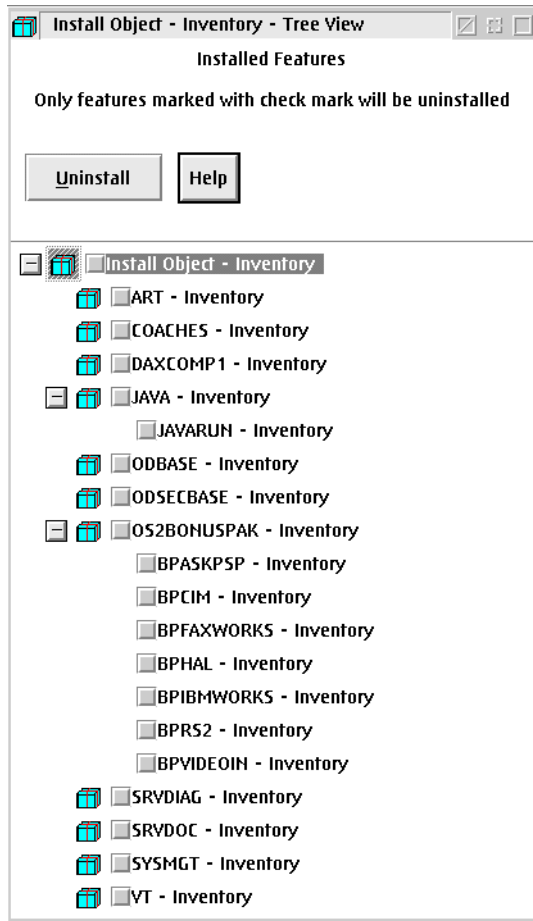


Figure 43. Install Object - Inventory Window

All features installed through ASD will be placed here and represented as an object. Uninstalling a feature is quite simple. Just mark the object's checkbox and click on the **Uninstall** button. For example, if you want to remove the OS/2 Warp 4 registration tool called Art, mark the representing object's checkbox and press the **Uninstall** button.

4.17.1 ASD-Allowed and ASD-PreReq Service and Support Issues

ASD-PreReq and ASD-Allowed features have dependencies on previously shipped system components:

- The dependencies must be met if they are to be successfully installed.

- ASD-Allowed features may update other components when their dependencies are met.

Installation and service tool inadequacies:

- Manage base module replacement when ASD features are installed and removed.
- Coordinate base module replacement by ASD features with service application and backout.

System characterization problems:

- System includes modules shipped with release, new modules installed by ASD features, modules applied as service, and prerequisite updates for ASD features
- Need ability to determine state of software installed on a customer system

4.17.2 ASD Requirements

Following requirements must be met by ASD:

- Service and support personnel can characterize the state of software installed on a customer's system:
 - What has been installed
 - Service level of installed files
 - Whether software dependencies are met
- ASD features can be installed and removed in any order
 - Locate prerequisites
 - Maintain backups of updated system modules
 - File-level version checking during install/removal of system modules
 - Handle corequisites
- Base updates during feature installation are safe and consistent with OS/2 maintenance tools
 - Installation failures do not render system unbootable or unserviceable
 - Maintain integrity of fixtool archives and backups
 - System updates applied during feature installation can be safely backed out when feature is removed

4.17.2.1 Solution

The solution is an application installation and removal algorithm that safely manages prerequisite module updates based on module versioning architecture.

Algorithm:

- Package metadata with application
- Locate prerequisite components on target system
- Archive versions of updated modules
- Backup modules that do not need to be updated
- Intelligent uninstall procedure
- Installation inventory to integrate archived modules with standard service tools

4.17.3 Design Assertions

In order to prevent ASD, ServicePaks, FixPaks (Fixtools) and features installed through Feature Installer from interfering with each other, a new design was created to reflect this. Fixtools and Feature Installer backups are synchronized when service is applied, backed out, or committed by Fixtools.

- Feature Installer does not maintain or modify Fixtool backups or logs.
- New version of Fixtool to perform synchronization.
- Old versions of Fixtool are prevented from servicing the system.
- Synchronization of Fixtool and Feature Installer backups commits updates to system modules made by Feature Installer.
- Fixtool will not downlevel a module installed with an ASD feature when applying service. All modules shipped with ASD-Allowed features have a new extended version string that contains a date/time stamp and three version-level fields:
 - A numeric product version level (value can be 1 to 255)
 - A numeric build-level number (value can be 1 to 2047)
 - An optional SubBuild number (value can be 1 to 32)
- Feature Installer and Fixtool will both examine all three fields when replacing modules on a target system. A higher number always means a later version for each version-level field.
- After FI base updates for ASD-Allowed features are synchronized and owned by Fixtool, they will not be backed out when features are removed.

- ASD backups and archives are stored under \OS2\INSTALL\ASD.

4.18 Feature Installer (FI)

Feature Install offers a set of installation services available to software developers which frees the software developer from writing customized installation code to install your software. In addition, for every install service used, Feature Install automatically supplies uninstall services.

The set of installation services includes file copies, updates to any ASCII file, and update to any .INI file. A software product can be broken down into manageable parts that can be selectively installed by the software consumer. Also, if there are important relationships between these various parts, there are services that run condition checks and perform specific actions. Finally, if there is a needed service that is not readily provided, there is the ability to run programs supplied by the software developer.

By using Feature Install to install your software, the software developer can standardize the installation process for their customers. This is particularly useful if you produce more than one product. Feature Install also provides the software developer the ability to customize the final product by providing a totally authorable view of your software product and override capabilities for important help panels.

4.18.1 Using the Command Line Interface (CLIFI)

The `CLIFI` command enables you to perform various Feature Install tasks using a command line interface. To use `CLIFI`, type `CLIFI` at an OS/2 command prompt.

The options are defined as follows:

- `/A:` Action code
 Following action codes are selectable and the following requirements apply when using these actions:
- `B` Build
- Requires `/R` (`/L1`, `/L2` are optional).
- `C` CID Install
- Requires `/R`, `/R2`, and `/S` (`/L1` and `/L2` are optional).
- `D` Delete Feature

I **Install**

Requires /O (/F, /L1, /L2 are optional) or /R (/L1, /L2 are optional).

O **Open Feature ID view**

Requires /O (/F, /L1, /L2 are optional).

/O should always be set to /O:"INV_xxx" (where xxx is the feature ID of the original install object) because inventory objects have an "INV_" prefix.

P **Package**

Requires /O (/F, /L1, /L2 are optional) or /R (/L1, /L2 are optional). Note that packaging can only be executed on the top install object.

T **Template object**

W **Write response file**

requires /O and /R (/F, /L1, /L2 are optional).

U **Uninstall**

Requires /O (/F, /L1, /L2 are optional).

If you use the /F option, specify /F:"<WP_INSTALLED>" because all inventory objects are located in that folder

/B: Boot drive with drive letter and colon (:)

/R: Drive, path, and file name of response file. This option must be specified with actions B, C, and W.

It is optional with actions I and P.

If a response file is not specified with actions I and P an object ID (/O) must be specified to perform I or P actions.

If a response file (/R) and an object ID (/O) are specified using the same CLIFI command for actions I and P then /O will be ignored.

/R2: Drive, path, and file name of partial response file.

This option is valid only with action C (CID install).

/S: Fully qualified path to installation media.

This option is valid only with action **C** (CID install).

/O: Feature ID to perform action on existing object.

This parameter must be specified with actions **D**, **O**, **W**, and **U**.

It is optional with actions **I** and **R**.

/F: Folder containing Feature Install object.

The default is `<WP_DESKTOP>`.

This parameter can be used with every action parameter.

With actions **B** and **T**, **/F** specifies the destination folder for newly created objects. If **/F** is not specified, an object is created on the desktop `<WP_DESKTOP>`.

For the following actions (together with **/O**): **D**, **O**, **W**, **U**, **I**, and **R**, **/F** specifies the location to start the search for the object ID.

If **/F** is not specified, **CLIFI** will set the search start location to `<WP_DESKTOP>`.

If the **/F** parameter is an object, such as `<WP_DESKTOP>`, or a path name containing spaces or unusual characters, enclose the object or directory name in double-quotation marks.

/REG: Specifies DLL to register. This parameter must be specified with **/CLASSNAME** (**/L1** and **/L2** are optional).

/CLASSNAME:

Specifies object class name to register. This parameter must be specified with **/REG** (**/L1** and **/L2** are optional).

/SET: Sets a variable (or response file keyword) for a given object. This parameter must be specified with **/O** (**/E**, **/L1**, **/L2** are optional).

/PARMS:

Drive, path, and file name of a parameters file, which is a file that contains **CLIFI** parameters. Each **CLIFI** parameter must be on a separate line.

For example, the **CLIFI** parameter file for the

`CLIFI /A:I /R:C:\CLIRSP.FIL`

command would contain the following:

```
/A:I
/R:C:\CLIRSP.FIL

/L1: Drive, path, and file name of error log file.
/L2: Drive, path, and file name of history log file.
```

The following examples demonstrate how to use the single-line CLIFI command:

```
CLIFI /A:I /R:\APPS\WIDGET.RSP /L1:D:\ERR.LOG /L2:D:\HIST.LOG
CLIFI /A:P /O:NewViewInc /F:"<WP_NOWHERE>"
CLIFI /A:W /O:NewViewInc /F:C:\OS2\INSTALL /R:RESPNEW.RSP
CLIFI /O:NewViewInc /SET:MediaSet[0].Media[0].MediaPath=C:\
CLIFI /REG:C:\OS2\DLL\INSTALL.DLL /CLASSNAME:WPInstall
CLIFI /A:C /S:Z:\OS2IMAGE /B:C: /R:C:\OS2\INSTALL\FIBASE.RSP
/R2:C:\OS2\INSTALL\SAMPLE.RSP /F:C:\OS2\INSTALL
/L1:Z:\LOGS\CLERR.LOG /L2:Z:\LOGS\CLHIST.LOG
```

Another CLIFI example is illustrated in Figure 95 on page 372.

Note: If packaging in an unattended environment using CLIFI, the media directory must be empty. If it is not empty, Feature Install displays a confirmation dialog asking permission to erase the directory.

Find another CLIFI execution example in "The Command Line Interface Feature Installer Change Profile" on page 371.

Chapter 5. Loading Product Images to the Code Server

This chapter illustrates the transfer of product images to the code server. Products discussed are:

- OS/2 Warp 4 - Base Operating System
- OS/2 Warp 4 - Networking Components
- eNetwork Communications Server for OS/2 Warp
- eNetwork Personal Communications for OS/2 Warp
- Database 2 for OS/2

5.1 Loading OS/2 Warp 4 - Base Operating System

OS/2 Warp 4 is shipped as a CD-ROM version only. Using the `XCOPY` command, transfer the CD-ROM's `\OS2IMAGE` directory onto the code server:

```
XCOPY E:\OS2IMAGE D:\CID\IMG\OS2WARP4 /S /E
```

assuming E: is the CD-ROM's driver letter and D:\CID\IMG\OS2WARP4 is the code server's directory. The `/S` parameter reflects subdirectories, and the `/E` parameter reflects empty subdirectories to be created.

5.2 Loading OS/2 Warp 4 - Networking Components

All OS/2 Warp 4 networking components reside on the CD-ROM's `\CID\IMG` directory. Simply copy all its subdirectories onto the code server:

```
XCOPY E:\CID\IMG D:\CID\IMG /S /E
```

We recommend renaming the `TCPAPPS` directory to `TCPIP`:

```
REN D:\CID\IMG\TCPAPPS TCPIP
```

5.2.1 MPTS Refresh Paks

MPTS is part of the OS/2 Warp 4 CD-ROM and is copied to the code server by using the above shown `XCOPY` command. Code refreshes of MPTS, sometimes referred as MPTS ServicePaks or FixPaks, however, come on diskettes and can be transferred to the code server using the `LAPSDISK` command which resides on the first MPTS diskette.

5.2.1.1 The LAPSDISK Command

LAPSDISK is a utility that allows copying the image of the first four MPTS diskettes to the code server. LAPSDISK executes XCOPY command.

LAPSDISK Syntax

```
LAPSDISK <source path> <target path>
```

<source path> Fully qualified source path

This specifies the source drive for the MPTS diskettes.

<target path> Fully qualified target path

Specifies MPTS target drive and path on the code server.

LAPSDISK Example

```
A:\LAPSDISK A:\ D:\CID\IMG\MPTS
```

5.2.2 Loading MPTS Utilities

The \CID\IMG\MPTS\UTILITY directory on the OS/2 Warp 4 CD-ROM provides additional utilities as well as sample MPTS response files. Refer to the README.UTL file for further information on these utilities in addition to the ones provided in this redbook. If you have MPTS diskettes, you can find these utilities along with the README.UTL file on Diskette 5.

You can also find a \SAMPLE directory which stores zipped MPTS response files. The file name is SAMPLE.ZIP. In addition to the LAPSRSP command, which creates MPTS response files out of PROTOCOL.INI files you can make use of these sample files by unpacking the file with the PKUNZIP2 command:

Unpacking SAMPLE.ZIP

```
PKUNZIP2 A:\SAMPLE\SAMPLE.ZIP D:\CID\RSP\MPTS
```

assuming A: is the source drive, and D:\CID\RSP\MPTS is the code server's location for MPTS response files

5.3 Loading the Mini Server SrvIFS

In case your software distribution manager being used is LAN CID Utility, you need to copy SrvIFS-related files from the OS/2 Warp CD-ROM to your code server:

```
XCOPY E:\CID\SRVIFS D:\CID\IMG\SRVIFS
```

5.4 Loading eNetwork Communications Server for OS/2 Warp

The eNetwork Communications Server for OS/2 Warp product is shipped on CD-ROM only. To load the product images to the code server, use the `XCOPY` command:

Copying eNetwork Communications Server Images

```
XCOPY E:\SERVER\*.* D:\CID\IMG\CS2 /S /E
```

Assuming that E: is the CD_ROM drive.

5.4.1 Communications Manager/2 Advanced Features

Use `XCOPY` to copy the `\CLIENTS\OS2\CM2AF` directory onto the CID structure

Assuming that E: is the CD_ROM drive:

Copying Advanced Features Images

```
XCOPY E:\CLIENTS\OS2\CM2AF\*.* D:\CID\IMG\CM2AF /S/E
```

5.4.2 Personal Communications Lite

Use `XCOPY` to copy the `\PCOMLITE` directory onto the CID structure

Assuming that E: is the CD_ROM drive:

Copying PCOMLITE Images

```
XCOPY E:\PCOMLITE\*.* D:\CID\IMG\PCOMLITE\ /S/E
```

5.5 Loading eNetwork Personal Communications for OS/2 Warp

Depending on the media of the software, perform the appropriate step.

5.5.1 Copying from Diskettes

Use `XCOPY` to copy each of the set of diskettes.

```
XCOPY A: D:\CID\IMG\PCOMOS2 /S /E
```

5.5.2 Copying from CD-ROM

If you purchased eNetwork Personal Communications for Windows, Windows 95, Windows NT, and OS/2 Warp, a CD-ROM is delivered with the package rather than product diskettes. To copy eNetwork Personal Communications for OS/2 Warp to the code server, use the following `XCOPY` command:

```
XCOPY E:\OS2\INSTALL\EMULATOR D:\CID\IMG\PCOMOS2 /S /E
```

We do not recommend copying the `E:\OS2\INSTALL\MPTS` product code to the code server since this MPTS version is lower than the one that comes with OS/2 Warp 4.

5.6 Loading Database 2 for OS/2

There is an diskette-based version of Database 2 for OS/2 available as well as a CD-ROM version.

5.6.1 Copying from Diskettes

There is no special CID utility to load the Database 2 for OS/2 diskette images to the code server. Use `XCOPY` with options `/S` and `/E` to copy the contents of the diskettes to the code server's hard disk.

All diskettes that belong to a single DB2 product are copied into a single directory. If the product consists of more than one diskette, you have to repeat the `XCOPY` command until all diskettes are copied.

Basic Rules

It is important to keep the files of the different DB2 products in separate directories.

You need directories for

- DB2 Server
- DB2 Single User
- DB2 Client Application Enabler for OS/2
- DB2 Software Developer's Toolkit
- Other DB2-related products, for example DDCCS/2

If you are using different DB2 products, create a parent directory DB2V211 with subdirectories for the different products:

DB2SRVR
DB2SU
DB2CAE2

Copying DB2/2 from Diskette

Copying DB2/2 from diskettes:

```
XCOPY A: D:\CID\IMG\DB2V211\xxx /S /E
```

where xxx is a name for the subdirectory of the product being copied.

5.6.2 Copying from CD-ROM

Use XCOPY to copy the code to the CID server.

The rules about directories introduced in 5.6.1 apply here, too.

In the root directory of your CD-ROM drive, look for a directory with a two-letter name that is identical with your country's language code. For example:

\EN for English
\GR for German

Copying DB/2 from CD-ROM

```
XCOPY E:\EN D:\CID\IMG\DB2V211\xxx /S /E
```

where xxx is a name for the subdirectory of the product being copied.

5.7 Loading OS/2 Warp Server File and Print Product Images

If you plan to distribute the OS/2 LAN Requester portion of OS/2 Warp Server, have the OS/2 Warp Server (Entry, Advanced, or SMP) CD-ROM handy and perform the following XCOPY commands to copy the product images to the code server:

```
XCOPY E:\CID\SERVER\IBMLS\IBM500R1 D:\CID\IMG\IBMLS /S /E  
XCOPY E:\CID\SERVER\IBMLS\IBM500R2 D:\CID\IMG\IBMLS /S /E  
XCOPY E:\CID\SERVER\IBMLS\IBM500R3 D:\CID\IMG\IBMLS /S /E  
XCOPY E:\CID\SERVER\IBMLS\IBM500R4 D:\CID\IMG\IBMLS /S /E  
XCOPY E:\CID\SERVER\IBMLS\IBM500R5 D:\CID\IMG\IBMLS /S /E
```

```
XCOPY E:\CID\SERVER\IBMLS D:\CID\IMG\IBMLS
```

It is not necessary to copy the the IBM500N1 to IBM500N5 directories because they represent a back-leveled MPTS version in comparison to the one in OS/2 Warp 4.

5.8 Loading CICS Images from CD-ROM

CICS transaction server images are loaded from the country subdirectory of the CD-ROM. It will be used either to install a server or a client.

In the root directory of your CD-ROM drive, look for a directory with a two-letter name that is identical with your country's language code. For example:

```
\EN for English  
\GR for German
```

Copying CICS from CD-ROM

```
XCOPY E:\EN D:\CID\IMG\CICS /S /E
```

5.9 Loading MQSeries from CD-ROM

Use `XCOPY` to copy the MQSeries code from CD-ROM to your CID server.

The easiest way is to copy all of the CD-ROM. In order to gain room on the hard disk, it is possible to copy all but the `\BOOKS` and `\SHELVES` directories.

If so, the `XCOPY` commands would look like:

Copying MQSeries images

```
XCOPY E:\AIX\*.*      D:\CID\IMG\MQ201\AIX /S /E  
XCOPY E:\DOS\*.*     D:\CID\IMG\MQ201\DOS /S /E  
XCOPY E:\EZ2INST\*.* D:\CID\IMG\MQ201\EZ2INST /S /E  
XCOPY E:\ILRWIN\*.*  D:\CID\IMG\MQ201\ILRWIN /S /E  
XCOPY E:\SCRT\*.*    D:\CID\IMG\MQ201\SCRT /S /E  
XCOPY E:\TAPES\*.*   D:\CID\IMG\MQ201\TAPES /S /E  
XCOPY E:\UPLOAD\*.*  D:\CID\IMG\MQ201\UPLOAD /S /E  
XCOPY E:\WIN\*.*     D:\CID\IMG\MQ201\WIN /S /E
```

5.10 Loading NetView DM/2 Diskette Images

The `NVDMCOPY` utility provided by NetView DM/2 copies all NetView DM/2 files to the specified target directory.

1. Create a proper subdirectory on your server.
2. Insert the NetView DM/2 Diskette 1 into drive A: and execute the following command:

```
NVDMCOPY /S:A: /T:D:\CID\IMG\NVDM21
```

If you want to load also the diskette images for the feature Remote Administrator, add the parameter `/RA` to the command `NVDMCOPY`.

3. The latest NetView DM/2 refresh is XR20466A as of March 1996; this is equivalent to `SYSLEVEL XR00002`. FixPak XREFP01, changing the `SYSLEVEL` to `XR00003`, is also recommended. To apply the CSD to the images on the server, insert the NetView DM/2 V2.1 Diskette 2 into drive A: and execute the following command:

```
A:\NVDMPDSD
```

You will get a Please wait while the Corrective Service Facility inspects the system message. Select **OK**.

The Corrective Service Facility menu will appear with a list of NetView DM/2 features and images that are installed on your hard disk:

Select all entries listed on the menu.

Select **SERVICE** to start the service process.

Insert each CSD diskette as prompted.

Note: `NVDMPDSD` can be used to apply the CSD to a NetView DM/2 image and/or the code installed on a hard disk.

5.11 Loading Lotus Notes OS/2 Client Product Images

To copy the Lotus Notes 4.51 OS/2 client images to the code server, insert the CD-ROM into your CD-ROM drive and perform the following `XCOPY` command:

```
XCOPY E:\OS2\INSTALL D:\CID\IMG\LOTUS\nOTES /S /E
```

Note: There is a `RESPONSE.RSP` response file on the CD-ROM's `\OS2\INSTALL` directory which should be used for a CID remote installation.

5.12 Loading Lotus SmartSuite 96 Product Images

To copy the Lotus SmartSuite 96 product images to the code server, insert the CD-ROM into your CD-ROM drive and perform the following `XCOPY` command:

```
XCOPY E:\PRODUCT D:\CID\IMG\LOTUS\SSUITE96 /S /E
```

Note: There is a SUITE.RSP response file on the CD-ROM's root directory which should be used for a CID remote installation.

Chapter 6. Remote Installation Commands for Additional Products

The following chapter presents the syntax information of remote installation programs for:

- eNetwork Communications Server for OS/2 Warp
- eNetwork Personal Communications for OS/2 Warp
- Database 2 for OS/2
- MQ Series for OS/2
- Transaction Server for OS/2 (CICS/2)

6.1 eNetwork Communications Server for OS/2 Warp

The eNetwork Communications Server for OS/2 Warp product uses the same installation program, `CMSETUP`, for both attended interactive configuration and installation as well as redirected response-file-driven installation.

Here, we will briefly explain how `CMSETUP` is invoked when doing a response file driven installation from a redirected drive.

Please note that since `CMSETUP` is an OS/2 PM program, even if it is called with parameters, it must be invoked from a fully functional OS/2 system.

This means that if you are using a software distribution manager to chain together installations, you have to ensure that the client is rebooted after the installation of OS/2, before it continues with the Communication Server installation.

Note: During the installation, `CMSETUP` uses temporary space on the client's boot drive. Ensure that the client has enough free space for this; otherwise the installation will break.

CMSETUP Syntax For Response File Driven

```
CMSETUP /R:<response file>
        /S:<source path>
        /G:<general path>
        /L1:<log file 1>
        /L2:<log file 2>
        /CR
        /MG:<migration file>
        /KL:<password>
        /Q
```

You must enter either a colon or a space after the parameters. You do not need to enter the file extensions.

- /R:** Fully qualified path and name of response file
- The response file name must have an .RSP extension (which can be omitted when `CMSETUP` is invoked).
- /S:** Fully qualified path to product images
- /G:** Fully qualified path
- Specifies the path to a directory on the code server that can contain a general response file or other data files. You may not specify a diskette drive.
- /L1:** Fully qualified path and log file name
- Specifies the fully qualified name of the file into which the log file for remote installation and configuration is to be copied.
- /L2:** Fully qualified path and log file name
- The installation log file.
- /KL:** Key lock password for configuration file
- /CR:** Current response file is made for Communication Server
- No check will be made to determine if it contains previous versions' specific keywords that need to be converted. If this parameter is not specified, the entire response file is checked to determine the level of the keywords. If they are the current level, remote installation or configuration continues normally.
- /MG:** Migration file name
- Indicates that the response file will be migrated and that the migrated response file should be saved to this name upon completion of the remote installation/configuration request. The path, if not specified, defaults to `\CMLIB`.
- This parameter is only used when you are migrating from a previous version response file and you want to save the output of the migration step. If you do not specify a migration file name, the default file name, `<rspfile>.MIG`, is used (where `<rspfile>` is the name of your response file).

/Q 'Quiet mode' no progress or message windows are shown.

CMSETUP.EXE must be invoked from the directory where the Communication Server diskette images are located on the code server. So CMSETUP does not need /S to be explicitly set.

CMSETUP Example

```
X:\IMG\COMSRVR\CMSETUP /CR
                        /L1 L:\COMSRVR\CMRINST.LOG
                        /L2 L:\COMSRVR\CMAUDIT.LOG
                        /R X:\RSP\COMSRVR\CLIENT1.RSP
```

CMSETUP is invoked from the redirected drive X:\IMG\COMSRVR. The response file, CLIENT1.RSP is a response file made for Communication Server. The log files will be logged on the redirected drive, L:\COMSRVR.

```
x.comsrvr = 8
x.8.name='COM Server'
x.8.statevar = 'CAS_' || x.8.name
x.8.instprog = 'X:\CID\IMG\COM_SRVR\CMSETUP ',
              '/CR ',
              '/L1:Z:\COM_SRVR\' || client || '.L1 ',
              '/L2:Z:\COM_SRVR\' || client || '.L2 ',
              '/r '
x.8.rspdir = 'X:\CID\RSP\COM_SRVR'
x.8.default = 'DEFAULT.RSP'
```

Figure 44. CMSETUP Program Invocation out of LCU

Be sure to reboot the client after CMSETUP is executed so that locked files are processed.

6.2 eNetwork Personal Communications for OS/2 Warp

The eNetwork Personal Communications for OS/2 Warp product is only used for the emulation functions and fundamental APPC communications support. All gateway functions are implemented in eNetwork Communications Server for OS/2 Warp. Here, we briefly explain the CID Installation of the product. The technical description is available in the online documentation of eNetwork Personal Communications for OS/2 Warp.

6.2.1 Base Product Installation

The INSTALL program is a PM program; so it has to be invoked from a fully functional OS/2 System. So there is at least one reboot needed after the installation of the base system to have the PM active.

Note: During the installation, the INSTALL program of eNetwork Personal Communications for OS/2 Warp uses temporary space on the client's boot drive. Ensure that the client has enough free space for this; otherwise the installation will not work.

PCOMOS2 INSTALL Syntax

```
INSTALL /R:<response file>  
        /A <centralized installation (server)>  
        /N <centralized installation (client)>  
        /S:<source path>  
        /G:<general path>  
        /T:<target path>  
        /L1:<errorlog file>  
        /L2:<historylog file>  
        /M:<type of communication stack>  
        /Q <quiet mode>
```

/A Centralized installation

Use this parameter to install eNetwork Personal Communications for OS/2 Warp in a network server from diskettes. This parameter does not create the PRIVATE subdirectory and does not set up the system settings.

/R: Fully qualified path and name of response file

The response file name must have an .RSP extension.

/S: Fully qualified source path

Specifies the drive and path of the product image files on the code server.

This parameter overrides the value specified by the keyword SOURCEPATH in the client-specific response file.

/G: Fully qualified general path

Specifies the drive and path of the general response files.

A general response file is referred to by an `INCLUDE` keyword in a specific response file

`/T:` Fully qualified target path

Specifies the target drive and path for the installation.

This parameter overrides the value specified by the keyword `TARGETPATH` in the client-specific response file.

`/L1:` Complete filename of the error log

Specifies the complete drive, path and filename for the error log file for this installation.

`/L2:` Complete filename of the history log

Specifies the complete drive, path and filename for the history log file for this installation.

`/M:` Kind of communication stack

When used along with the `/R:` parameter, specifies the target communication stack to be used for CID migration.

If `/M:S`, the migration is to stand-alone eNetwork Personal Communications for OS/2 Warp.

If `/M:C`, the migration is to eNetwork Personal Communications for OS/2 Warp using CM/2 communication.

`/N` Centralized installation

Use this parameter when installing eNetwork Personal Communications for OS/2 Warp in a network server using the `/A` parameter and when the installed programs must be shared by the client workstations.

The `PRIVATE` subdirectory is created, and the system settings are set up, but the program files are not copied to the target directory.

`/Q` Quiet mode

In the quiet installation mode, the information windows are suppressed. If this parameter is omitted, there will be a prompt dialog waiting for an Enter key to be pressed to continue installation!

```

x.pcom = 8
x.8.name='Pcom OS/2'
x.8.statevar = 'CAS_' || x.8.name
x.8.instprog = 'x:\img\pcomos2\install.exe ',
              '/m:S ',
              '/s:x:\img\pcomos2 ',
              '/11:Z:\comm\' || rndnom || '.11 ',
              '/12:Z:\comm\' || rndnom || '.12 ',
              '/r:'
x.8.rspdir = 'x:\rsp\pcom'
x.8.default = 'poste.rsp'
The clients connected to this server can only use the emulation that was
chosen for the server. To get access to this

```

Figure 45. PCOMOS2 Install Program Invocation out of LCU

PCOM LITE Install

PCOM LITE, included with eNetwork Communications Server for OS/2 Warp, is to be installed exactly as eNetwork Personal Communications for OS/2 Warp.

6.2.2 Advanced Features Installation

Advanced features add the APPC support to eNetwork Personal Communications for OS/2 Warp. It is to be installed by the `CMSETUP` program. See Chapter 6.1, “eNetwork Communications Server for OS/2 Warp” on page 171, for information on `CMSETUP`.

```

x.cm2af = 20
x.20.name='CM Adv Features'
x.20.statevar = 'CAS_' || x.20.name
x.20.instprog = 'x:\cid\img\cm2af\cmsetup ',
              '/cr ',
              '/11:Z:\cm2af\' || client || '.11 ',
              '/12:Z:\cm2af\' || client || '.12 ',
              '/r '
x.20.rspdir = 'x:\rsp\cm2af'
x.20.default = 'default.rsp'

```

Figure 46. CM/2 Advanced Features Install out of LCU

6.3 Database 2 for OS/2

IBM Database 2 for OS/2 Version 2 uses Software Installer for installation. Many DB/2 products can be loaded this way. This is a list of the available products for the US version:

- IBM DB2 for OS2 - Server
- IBM DB2 for OS2 - Single User
- IBM DB2 Client Application Enabler for OS2
- IBM DB2 Software Developer's Kit for OS2
- IBM DB2 Administrator's Toolkit for OS2
- IBM DDCS Multi-User Gateway for OS2
- IBM DDCS Single-User for OS2

See 4.16.1, "Parameter Definitions" on page 151, for information on how to run the INSTALL program.

Option /x is required for unattended installation as well as the /R and /A parameters.

Figure 47 illustrates the LCU installation of DB2 Client Application Enabler (CAE). If the chosen product to install has several components, those to install have to be described in the response file.

```
x.db22 = 15
x.15.name='DB2/2 client '
x.15.statevar = 'CAS_' || x.15.name
x.15.instprog = 'x:\cid\img\db2v2\db2cae2\install.exe ',
               '/X /A:I ',
               '/P:"IBM DB2 Client Application Enabler for OS2" ',
               '/11:Z:\db2v2\' || client || '.11 ',
               '/12:Z:\db2v2\' || client || '.12 ',
               '/r:'
x.15.rspdir = 'x:\rsp\db2v2'
x.15.default = 'db2cae2.rsp'
```

Figure 47. Extract of an LCU Client Command File for DB/2 Install

6.4 MQ Series for OS/2

MQ Series uses Software Installer for installation.

See 4.16.1, "Parameter Definitions" on page 151, for information on how to run the INSTALL program.

Option /x is required for unattended installation as well as the /R and /A parameters.

Figure 48 shows an example of LCU installation of MQ Series.

```
x.mqseries = 18
x.18.name='MQ Series V2'
x.18.statevar = 'CAS_' || x.18.name
x.18.instprog = 'x:\img\mq201\install.exe ',
               '/X /A:I ',
               '/P:"IBM MQSeries for OS/2" ',
               '/l1:Z:\MQM\' || client || '.11 ',
               '/r:'
x.18.rspdir = 'x:\rsp\mqm'
x.18.default = 'mqm.rsp'
```

Figure 48. Extract of an LCU Client Command File for MQ Series Install

Installation of a MQSeries client is the same. Only the components are different as they should be indicated in the response file.

Note: The installation process installs the product as required. All the necessary definitions of channels, queues, and so forth have to be made in a separate customization process.

6.5 Transaction Server Installation (CICS)

The transaction server uses the Software Installer to install.

Figure 49 illustrates the LCU installation of CICS server, and Figure 50 illustrates the LCU installation of a CICS client.

See 4.16.1, "Parameter Definitions" on page 151, for information on how to run the INSTALL program.

Option /x is required for unattended installation as well as the /R and /A parameters.

```

x.cics = 1
x.1.name='CICS V3'
x.1.statevar = 'CAS_' || x.1.name
x.1.instprog = 'x:\cid\img\cics300\install ',
              '/X /A:I ',
              '/S:x:\cid\img\cics300 ',
              '/11:Z:\cics\' || client || '.11 ',
              '/12:Z:\cics\' || client || '.12 ',
              '/r:'
x.1.rspdir   = 'x:\cid\rsp\cics'
x.1.default  = 'cicssrv.rsp'

```

Figure 49. Extract of an LCU Client Command File for CICS Server Install

```

x.cics = 1
x.1.name='CICS V3 client'
x.1.statevar = 'CAS_' || x.1.name
x.1.instprog = 'x:\cid\img\cics300.cli\install ',
              '/X /A:I ',
              '/11:Z:\cics\' || client || '.11 ',
              '/12:Z:\cics\' || client || '.12 ',
              '/r:'
x.1.rspdir   = 'x:\cid\rsp\cics'
x.1.default  = 'cicscli.rsp'

```

Figure 50. Extract of an LCU Client Command File for CICS Client Install

6.6 Lotus Notes Remote Installation

Lotus products use Software Installer to install either locally or remotely. The following is the syntax of the installation program:

INSTPM.EXE Syntax

```

<path>\INSTPM.EXE /X
                /R:<responsefilepath>
                /P:"<installtype>"
                /L2:<logfilepath>

```

where:

<path> The path to the Notes install program executable

/X Turns on the automated install feature
 /R:<responsefilepath> Specifies the full path to the response file
 /P:"<installtype>" Specifies an install type as follows:
 /P:"Lotus Notes for OS/2 Workstation"
 /P:"Lotus Domino(tm) Server, Powered by Notes."
 /P:"Lotus Notes for OS/2 Shared"
 /P:"Lotus Domino for OS/2 Advanced Services"
 /P:"Lotus Notes for OS/2 Node"

6.7 Lotus SmartSuite 96 Remote Installation

Lotus SmartSuite 96 uses Software Installer to install either locally or remotely. Following is the syntax of the installation program:

INSTALL Syntax

```

<path>\INSTALL.EXE /X
                /A:I
                /L1:<logfilepath>
                /P:"<installtype>"
                /S:<source directory>
                /R:<responsefile>
  
```

<path> Is the fully qualified install program path
 /X Indicates unattended install
 /A:I Indicates to install
 /L1:<logfilepath> Is the fully qualified log file path
 /S: Specifies the fully qualified path to the source directory
 /R: Specifies the fully qualified path and response file name
 /P:"<installtype>" Specifies an install type as follows:
 /P:"Freelance Graphics for OS/2"
 /P:"Word Pro for OS/2"
 /P:"1-2-3 and Freelance File Translator"
 /P:"SmartCenter 2.0"

Chapter 7. Creating Response Files

This chapter provides information about product-specific response files, such as:

- OS/2 Warp 4 Base Operating System along with Feature Installer
- MPTS
- OS/2 Peer Services
- OS/2 LAN Requester
- TCP/IP
- Remote Access Services (LAN Distance)
- NetFinity
- SystemView Agent
- Network SignOn Coordinator
- Mobile FileSync (MFS)
- Software Installer
- NetView DM/2 CC Agent
- Database 2 for OS/2
- MQ Series
- Transaction Server (CICS)
- Feature Installer Partial Response File
- eNetwork Communications Server for OS/2 Warp
- eNetwork Personal Communications for OS/2 Warp

We will show how to create these response files where appropriate and display the response files used in this redbook. In addition, remote installation programs are illustrated where appropriate.

The standard installation process requires inserting diskettes and answering screen prompts to provide configuration information. When response files are used, all information necessary for the installation is provided by these files. The response file interface works in a batch-oriented fashion. It effectively turns off the dialog-oriented user interface. In some cases, progress indicators are displayed during an unattended install.

In their purest form, the response files used in the CID installation process replace the required human intervention that must take place during a fully attended installation of OS/2 and any associated products on a client workstation. The response files are ASCII files that contain general as well as client-specific configuration information that is understood by the CID installation process. Response files allow for an installation process that can be either lightly attended or completely unattended. Response files used in the CID installation process commonly have an extension of .RSP and are found in the \RSP subdirectories under the CID root directory.

The code server administrator has to build the response files in order to install products on client workstations. This chapter covers how to create response files for each product and points out if there is anything unique in the way the response file is interpreted by a given product.

7.1 OS/2 Warp 4 - Base Operating System

A sample response file called SAMPLE.RSP can be found in the \OS2\INSTALL directory of an installed system. Not all keywords in SAMPLE.RSP are actually documented. In "Detailed Keyword Description" on page 185 there is a detailed description of all keywords.

SAMPLE.RSP can be copied and edited by the administrator to tailor the needs of individual client workstations. The administrator can create a default response file for all client workstations or create an individual response file for each client workstation.

An OS/2 response file contains two keywords called `ExitOnError` and `RebootRequired` that are mandatory for a successful CID installation of OS/2. These two keywords must be changed and set to the following values:

Necessary Values

```
ExitOnError = 1
RebootRequired = 0
```

When using nested include files with the OS/2 response file, it should be noted that if a number of levels of nested include files are used, the parameters of the last include file will override those of higher-level include files. Care should be taken when using nested include files with the OS/2 response file.

Please read through the explanations of the keywords carefully and review the settings so you really understand what the installation program will install with a particular response file. Some of the more important keywords to understand are listed here:

`FormatFAT` Specify the hard disk drive letters that should be formatted using the FAT file system.

`FormatHPFS` Specify the hard disk drive letters that should be formatted using the HPFS file system.

Support for Old Keywords

Use the new keywords

FormatFAT
FormatHPFS

for installation of OS/2 Warp 4. Although the older keywords

BaseFileSystem
FormatPartition

are still supported, you should not use them any longer.

Please note, that the old and new sets of formatting keywords are mutually exclusive. So, don't mix them in a single response file!

CountryCode	If you are using a national language version, this usually defaults to the correct country code.
CountryKeyboard	Check that this matches the country keyboard you will install for.
PrimaryCodePage	For most countries (at least those using Single Byte Character Set), it should be set to 'Multilingual' (which is the default for many national versions).
	If the country settings is not correct and Windows support is to be installed this might not install the correct country settings either. Which means that all of them has to be changed afterwards by the user. To discover this, may take some time before the user runs some Windows applications and is not getting the expected behavior. So ensure it is correct from the beginning!
DefaultPrinter	Detailed information about this parameter is provided in the "Printer Description Table PRDESC.LST" on page 209. It is wise to test that the choice you make really installs the printer you intended to use.
DisplayAdapter	Even though it is possible to specify "SVGA support", this will not lead to a full install of SVGA because it is not possible to specify the type of video adapter or the resolution during the install.

There is a possibility to do the settings by running a separate program after the OS/2 installation (DPSINSTL).

In case you are not certain about the display adapter to be used, specify VGA.

MultimediaSupport This keyword, if set to 1, will install multimedia files and programs. However, the multimedia configuration, such as sound card configuration, is done by a separate procedure during the OS/2 installation.

Values used for installation can be viewed after installation either on the code server in the \CID\LOG\OS2WARP4 log file directory or in the INSTALL.LOG file in the \OS2\INSTALL directory at the workstation. For keywords not specified, default values are used. Here is an abstract of an INSTALL.LOG file:

```
Processing: TargetDrive=C:
Processing: ToolsAndGames=1
Processing: ConfigSysLine=SET RESTARTOBJECTS=STARTUPFOLDERONLY
Processing default: Copy=
Processing default: EarlyUserExit=
Processing default: ExtendedInstall=
Processing default: Include=
Processing default: OptionalSystemComponents=1
Processing default: MigrateApplications=
```

All keywords described in “New Keywords in OS/2 Warp 4” on page 204, appear in the INTALL.LOG file, but they are not installed through SEINST/RSPINST. The corresponding products are installed by Feature Installer (CLIFI).

7.1.1 Use of Client-Specific Response File for OS/2 Installation

In many environments, one default response file, named for example DEFAULT.RSP, could be used for most installations. In cases of a client that needs special configuration settings, you can provide a client-specific response file in which you include the `IncludeInLine` keyword specifying the default response file. Then for those keywords that differ, define them to whatever values they should be.

For example, if the DEFAULT.RSP file specifies the install drive to be the C: drive, and one client should be installed on drive D:, the response file for this client should look like:

```
IncludeInLine = X:\RSP\OS2WARP4\DEFAULT.RSP
TargetDrive=D:
```


If your software distribution manager being used is NetView DM/2 V2.1, you may want to use client-specific response files in order to keep track of what really is installed on the client.

7.1.2 Detailed Keyword Description

The following is a description of all the keywords and valid entries that can be used in a OS/2 Warp 4 response file. Some keywords used in previous versions of OS/2 are not supported any more. The new keywords are described in "New Keywords in OS/2 Warp 4" on page 204.

- **AdditionalPrinters**

Allows additional printers other than the default printer to be installed.

Key values are 0(=None) or m1:n1,m2:n2,...

Where m1, m2, and so on are port numbers as defined under "PrinterPort" below, and n1, n2, and so forth are indexes into PRDESC.LST as described in "Printer Description Table PRDESC.LST" on page 209.

Example:

```
AdditionalPrinter=2:256
```

This will put an IBM 2380 PPS II on LPT2: The 256 specifies the PPS II (if it is line 256 in PRDESC.LST) and the 2 specifies LPT2.

You can use multiple port:printer specifications on this line. You can also have multiple `AdditionalPrinters` statements.

- **AlternateAdapter**

Specifies secondary adapter for two display systems. This should be a lower or equal resolution display since the highest resolution display will be primary for Presentation Manager.

```
0 = None (DEFAULT)
1 = Other than following (DDINSTAL will handle)
2 = Monochrome/Printer Adapter
3 = Color Graphics Adapter
4 = Enhanced Graphics Adapter
5 = PS/2 Display Adapter
6 = Video Graphics Adapter
7 = 8514/A Adapter
8 = XGA Adapter
9 = SVGA Adapter
```

- **APM (Advance Power Management)**

Specifies whether or not to install APM.

```
0 = Don't install
```

1 = Autodetect (DEFAULT)
2 = Install

- **BaseFileSystem**

Specifies which file system should be used to format the install partition, for example, HPFS or FAT.

1 = HPFS(DEFAULT)
2 = FAT

This keyword cannot be used in conjunction with the `FormatFAT` or `FormatHPFS` keywords.

- **CDROM**

Specifies which, if any, CD ROM devices you wish to install support for.

The values that can be entered here are:

0 = None
1 = Autodetect
2=IDECD-ROM
3=SCSI CD-ROM
4=Acer525E,625A,6825P,743E,747E
5=AztechCDA268-03I
6=AztechCDA268-03I-SE
7=AztechCDA468-0II
8=AztechCDA468-02I
9=AztechCDA668-01E
10=AztechCDA668-01I-SE
11=AztechIDE
12=CDTechnology T3301, T3401
13=Chinon525I, 545I
14=Chinon431, 435, 535
15=CompaqIDE Dual Speed, Quad Speed, Tray Load
16=CompaqSCSI-2 Tray Load
17=CompaqSCSI-2 Dual Speed
18=CreativeLabs OmniCD
19=CreativeLabs IDE CD220E,CD420E,CD620E
20=EliteGroup Computer Systems VERTOS 300SSD Elite
21=GoldstarGCD-R520B,GCD-R540B,GCD-R560B,GCD-R580B
22=Hitachi7730,7930
23=Hitachi1650S,1750S,3650
24=Hitachi1950S,3750,6750
25=IBMIDE 2X,4X,6X CD-ROM
26=IBMCD-ROM I
27=IBMCD-ROM I rev 242
28=IBMCD-ROM II, Enhanced CD-ROM II
29=IBMISA,Panasonic 562,563
30=LionOptics XC-200AI,XC-200EI,XC-600EI

31=MitsumiCRMC-LU002S,Tandy CDR-1000
 32=MitsumiCRMC-LU005S
 33=MitsumiCRMC-FX001
 34=MitsumiCRMC-FX001D
 35=MitsumiCRMC-FX001DE,FX300,FX400,FX400E,FX410A,FX600S
 36=NECIntersect 25,36,37,72,73,74,82,83,84
 37=NECMultiSpin 4Xc,4Xe,4Xi,3Xi,3Xe,3Xp,38,74-1,84-1
 38=NEC2vi,250,260,260GW,260R,271,272,273,MultiSpin 6V,8V,4x4
 39=OpticsStorage 8001 IDE,8222 IDE,8322,8422
 40=PanasonicCF-41,CR-571,CR-572,CR-574,CR-581,LK-MC579
 41=PanasonicCR-501
 42=PanasonicLK-MC501S
 43=Panasonic521,522,523
 44=PhilipsLMS CM-205,CM-225
 45=PhilipsLMS CM-205MS,206,225MS,226
 46=PhilipsLMS CM-215
 47=PhilipsLMS CM-207
 48=PhilipsProfessional Solutions PCA62CR,PCA62CR 6x
 49=PhilipsIDE
 50=PioneerDRU-A124XP
 51=PioneerDRM-600
 52=PioneerDRM-604X
 53=PlextorDM-3028,DM-5028,4PLEX
 54=SanyoCRD-254P,CRD-450P
 55=SonyCDU-31A,33A,7305,7405
 56=SonyCDU-531,535,6150,6201,6205,6251,7201,7205
 57=SonyCDU-55D,55E,76E,760E,77E
 58=Sony541,561,6211,7211,7811
 59=Sony6111
 60=Texel5302B
 61=Texel3021,5021
 62=Texel3024,3028,5024,5028
 63=ThinkPad755CD, TEAC CD-40E,CD-56E,CD-58E
 64=Toshiba3101,3201
 65=Toshiba3301,3401,3501,3601,3701,4101,5201,5301,5401
 66=Toshiba5302B
 67=WearnesCDD-120,CDD-620,CDD-820

- **ConfigSysLine**

Specifies a text line to be appended to CONFIG.SYS. There may be multiple occurrences of this keyword. No validity checking is done; therefore, statements entered into the CONFIG.SYS must be correct.

Key value is any valid CONFIG.SYS statement, for example:

```
ConfigSysLine=SET RESTARTOBJECTS=STARTUPFOLDERONLY
ConfigSysLine=PAUSEONERROR=NO
ConfigSysLine=AUTOFAIL=YES
ConfigSysLine=SUPPRESSPOPUPS=C
ConfigSysLine=RESERVEDRIVELETTER=G
```

- **Copy**

Specifies a source file from either the client or the server to be copied to a destination directory on the client or server during the install process. Errors are ignored though they will be logged in the INSTALL.LOG file, in the install directory of the client C:\OS2\INSTALL. For example, there could be a copy statement that copies a file from the client to the server.

Copy statements are executed on completion of the installation of each diskette. The reason being that the user may not be sure when the file will be available to be copied, therefore repeating the copy after each diskette.

There may be multiple occurrences of this keyword. No validity checking is done.

The command issued is not the OS/2 COPY command, it is an UNPACK command. Therefore, the file that is being unpacked must be unpacked to its original name.

The key value is "*sourcefile destination*" where:

- *sourcefile* is a valid file name
- *destination* is a valid directory name

Example:

```
Copy = Z:\TEXT.DAT C:\
```

- **CountryCode**

Specifies which country code should be installed. This causes all country information to be installed. Use of this keyword will update the Country panel in System Setup, or the WIN-OS2 panel.

Country	Country	Codepage	Code
785	Arabic-speaking	864,	850
099	Asian English	437,	850
061	Australia	437,	850
032	Belgium	437,	850
055	Brazil	437,	850
002	Canada (French-speaking)	863,	850
042	Czechoslovakia	852,	850
045	Denmark	865,	850
358	Finland	437,	850
033	France	437,	850

049	Germany	437, 850
972	Hebrew-speaking	862, 850
036	Hungary	852, 850
354	Iceland	850, 861
039	Italy	437, 850
081	Japan	932
082	Korea	934
003	Latin America	437, 850
031	Netherlands	437, 850
047	Norway	865, 850
048	Poland	852, 850
351	Portugal	860, 850
086	China	936
034	Spain	437, 850
046	Sweden	437, 850
041	Switzerland	437, 850
088	Taiwan	938
090	Turkey	857, 850
044	United Kingdom	437, 850
001	United States	437, 850
038	Yugoslavia	852, 850

- **CountryKeyboard**

Specifies which country keyboard should be installed. This causes all keyboard information to be installed. It is a 2-5 character keyboard code.

AR	= Arabic
BE	= Belgium
BR	= Brazil
CF	= Canada,
CS243	= Czechos
CS245	= Czechos
DK	= Denmark
SU	= Finland
FR189	= France
FR120	= France,
GR	= Germany
HE	= Hebrew
HU	= Hungary
IS	= Iceland
IT141	= Italy
IT142	= Italy, Enhanced Keyboard
LA	= Latin America
NL	= Netherlands
NO	= Norway
PO	= Portugal
PL	= Poland

SP = Spain
SV = Sweden
SF = Switzerland, French
SG = Switzerland, German
TR = Turkey
UK166 = United Kingdom
UK168 = United Kingdom
US = United States
YU = Yugoslavia

- **DDIDDP**

Use the OS/2 Device Driver Installation program to install external loadable device drivers. A Device Driver Profile (a text file with a .DDP file name extension) must be provided by the device driver author to control the installation of the device driver.

The key value is a list of .DDP files to install.

Example:

```
DDIDDP=file1.DDP,file2.DDP
```

Note

Some CPUs with loadable BIOS need special files delivered with the hardware. For information, see Hardware and Software Dependencies. If the installation is to be done to such a system, the administrator should proceed as follows:

- Activate the system partition by pressing **ALT+CTRL+INS**, when the cursor is located in the upper-right corner of the display.
- Select **Backup/Restore of System Programs** from the first menu.
- Select **Backup Reference Diskette** and follow the displayed instructions.
- Reject the creation of the Diagnostic diskette (press the **Esc** key).
- Create a new subdirectory in the \CID\IMG\OSWARP4 directory on the code server.
- From the backup reference diskette copy BIOS.SYS, *.BIO and *.DDP files into that directory.
- Reference this directory by the `DDISrc` keyword.

- **DDIDest**

The OS/2 Device Driver Installation Destination. This determines the target directory for the device driver. (See also **DDIDDP** and **DDISrc**.)

The key value is a directory where to copy the device driver files.

- **DDISrc**

The OS/2 Device Driver Installation Source. This determines the source directory for the .DDP files. (See also **DDIDDP** and **DDIDest**.)

The key value is a directory where the .DDP files are:

Using DDInstall for BIOS for Micro Channel-based Machines

Using DDInstall, the response file has to have the following entries:

```
DDISrc   = Z:\IMG\OS2WARP4\ABIOS\PC700
DDIDest  = C:\
DDIDDP   = BIOS.DDP
```

Please keep in mind that you need a fully qualified path to the DDISrc, even when using NVDM/2. You can easily adopt the mechanism of DDInstall for other files; check an existing *.DDP file for the syntax used in it.

Be careful when using DDInstall because the OS/2 install will not fail if the DDISrc is not found, and therefore the DDInstall is not executed. However, your install might not work without the DDInstall, for example, if used for BIOS files on a system without a reference partition. Therefore, double-check the paths entered here, and check the file name for the DDP file.

- **DefaultPrinter**

Specifies which default printer to install.

The key value is the index in the PRDESC.LST file or 0 if none.

See "Printer Description Table PRDESC.LST" on page 209, for an explanation of how to find the key value.

- **DisplayAdapter**

Specifies which adapter should override the primary adapter detected by the install process.

```
0 = Accept as correct (DEFAULT)
1 = Other than following (DDINSTALL will handle)
2 = Color Graphics Adapter
3 = Enhanced Graphics Adapter
4 = Video Graphics Adapter
5 = 8514/A Adapter
6 = XGA Adapter
7 = SVGA Adapter
```

- **Documentation**

Specifies which documentation should be installed.

0 = None
1 = All (DEFAULT)
2 = OS/2 Command Reference
3 = OS/2 Tutorial
4 = REXX Documentation

- **DOSSupport**

Specifies whether or not to install DOS support under OS/2. If installed, it enables the use of Virtual DOS Machines (VDMs) under OS/2.

0 = Don't install DOS
1 = Install DOS (DEFAULT)

- **DPMI**

Specifies which DOS Protect Mode Interface options to install.

0 = None
1 = All (DEFAULT)
2 = Virtual DOS Protect Mode Interface
3 = Virtual Expanded Memory Management
4 = Virtual Extended Memory Support

- **EarlyUserExit**

Specifies the name of a program that Install will DosExec after the target drive is prepared. Install waits for the program to return. This keyword may occur more than once. Each will be executed in the order that they appear at the end of OS/2 Install. The only difference between this keyword and the `UserExit` keyword is that this one is executed early in the installation process while the latter is executed at the very end.

The key value is a user exit program name (DEFAULT=none)

For an example on the use of `EarlyUserExit` refer to "The User Exit Keywords of the Response File" on page 214.

- **ExistingWindowsPath**

Specifies the path to an existing Windows system. This option is valid only when option 1 is selected for the WIN-OS/2 desktop key value.

The key value is a string that specifies the path to the existing Windows system

Example:

```
ExistingWindowsPath=C:\WINDOWS
```

- **ExitOnError**

Specifies if the install program should exit with an error code if an error occurs. This also determines whether the installation process will exit with a return code when it completes rather than the Ctrl-Alt-Del panel.

0 = Do not exit when error occurs; display panel (DEFAULT)
1 = Exit quietly with a return code

Note

For CID installations, it is required that `ExitOnError` is set to 1.

- **ExtendedInstall**

Specifies .EXE or .CMD to be run. These programs were started from RSPINST.EXE by the DosExec API when the RSPINST application finalized the installation process and possibly ran the UserExit. There is no capture of return codes back to the RSPINST.EXE. The ExtendedInstall execution will be recorded to installation log file.

The key value is a full path name of program (DEFAULT=none).

- **Fonts**

Specifies which fonts should be installed.

0 = None
1 = All (DEFAULT)
2 = Courier (Bitmap)
3 = Helvetica (Bitmap)
4 = System Mono-spaced (Bitmap)
5 = Times Roman (Bitmap)
6 = Courier (Outline)
7 = Helvetica (Outline)
8 = Times New Roman (Outline)

- **FormatFAT**

Specifies which drives to format with the FAT file system. This keyword cannot be used in conjunction with the `BaseFileSystem` OR `FormatPartition` keywords.

The key value is drives to format as FAT.

Example:

`FormatFAT=C: ,D: ,E:`

- **FormatHPFS**

Specifies which drives to format with the HPFS file system. This keyword cannot be used in conjunction with the `BaseFileSystem` OR `FormatPartition` keywords.

The key value is drives to format as HPFS.

Example:

```
FormatHPFS=F: ,G:
```

- **FormatPartition**

Specifies whether or not to format the install partition by using the HPFS or FAT file system chosen in the `BaseFileSystem` keyword.

This keyword cannot be used in conjunction with the `FormatFAT` or `FormatHPFS` keywords.

0 = Do not format (DEFAULT)

1 = Format

- **Include**

Specifies another response file which will include additional keywords or override the current response files-keywords. Different include files could therefore be used for those specific workstations whose requirements are not met by a standard response file. If duplicate keywords appear, the last occurrence will be used. There can be multiple occurrences of this keyword.

Note

The included response files will always be processed after the main response file is interpreted.

The fully qualified path is required when specifying an include file.

The key value is a valid file name.

- **IncludeInLine**

Specifies another response file which will include additional keywords or override the current response files keywords. Different include files could therefore be used for those specific workstations whose requirements are not met by a standard response file. If duplicate keywords appear, the last occurrence will be used. There can be multiple occurrences of this keyword.

Note

The included response files will be processed in the order in which they appear in the main response file. The location of the include file plays a major role in determining the keywords value.

The fully qualified path is required when specifying an include file.

The key value is a valid file name.

- **Infrared**

Specifies whether or not to install Infrared Port Support

0=Don't install (DEFAULT)
1=Install

- **MigrateApplications**

Specifies whether or not to migrate existing DOS, Windows and OS/2 applications. Only those applications listed in the database specified will be migrated.

The key values are drives to search and databases to use for search.

Example:

MigrateApplications=C:D: ,C:\OS2\INSTALL\DATABASE.DAT

- **MigrateConfigFiles**

Specifies whether or not to migrate configuration files from a previous release of the operating system, thus allowing your existing CONFIG.SYS to be migrated. The AUTOEXEC.BAT for DOS will also be migrated.

0 = Don't migrate
1 = Migrate files (DEFAULT)

- **Mouse**

Specifies which mouse device driver, if any, to install.

0 = No pointing device support
1 = PS/2 Style Pointing Device (DEFAULT)
2 = Bus Version
3 = Serial Version
4 = InPort Version
5 = Logitech (tm) 'C' Series Serial Mouse
6 = IBM PS/2 Touch Display
7 = Logitech 'M' Series Mouse
8 = PC Mouse Systems (tm) Mouse
9 = Other Pointing Device for Mouse Port

- **MousePort**

Specifies to which port a serial-type mouse should be attached (valid for serial or Logitech mice).

0 = No port necessary (DEFAULT)
1 = COM1
2 = COM2

3 = COM3

4 = COM4

- **MultimediaSupport**

Specifies whether or not to install multimedia files during the installation. No configuration is supported at this point. For further information, see Multimedia Support.

0 = do NOT install multimedia support

1 = install multimedia support (DEFAULT)

Example:

MultimediaSupport=1

- **Optical**

Specifies whether or not to install SCSI -II Optical Device Support.

0=Don't install (DEFAULT)

1=Install

- **OptionalSystemComponents**

Specifies whether or not to install the following system components.

0=Install none

1=Install all (DEFAULT)

2=Truemode

3=HPFS

- **OptionalSystemUtilities**

Specifies whether or not to install the available system utilities.

0 = Install none

1 = Install all (DEFAULT)

2 = Backup Hard Disk

3 = Change File Attributes

4 = Display Directory Tree

5 = Manage Partitions

6 = Label Diskettes

7 = Link Object Modules

8 = Picture Utilities

9 = PMREXX

10 = Recover Files

11 = Restore Backed-up Files

12 = Sort Filter

13 = Installation Aid

14 = Create Utility Diskettes

15=Servicability and Diagnostic Aids

Example:

OptionalSystemUtilities=2,9,4

would install Backup, PMREXX and Tree utilities.

- **OS2IniData**

Specifies a profile string to be written to the user configuration file OS2.INI. There may be multiple occurrences of this keyword. This statement utilizes the PrfWriteProfileString API, defined in the *PM Programming Reference* in the chapter titled "Profile Functions". Valid keywords are found in the Appendix titled "Initialization File Information". It is possible to add private application names with private key names and key values, which can be retrieved by a private application using the PrfQueryProfileString API. Application names starting with "PM_" are reserved for Presentation Manager.

The key value is /AppName/KeyName/KeyValuE/.

Note

Since each of these names can contain imbedded blanks, the "slash" character must be used as a delimiter. There must be three tokens delineated on all sides, or this keyword will be ignored.

Example:

```
OS2IniData=/PM_SPOOLER/QUEUE/PSCRIPT2
```

This would define the default queue name.

- **PCMCIA**

Specifies whether or not to install PCMCIA.

Valid values are:

```
0=Don't install (DEFAULT)
1=Ambra486 SN425C
2=AmbraTS30AS
3=ASTAscentia 800N
4=ASTAscentia 900N
5=ASTBravo
6=ASTPowerExec
7=AustinBusiness Audio 4/75 Dual Scan
8=AustinDSTN
9=CompaqConcerto
10=CompaqContura,Contura 410,Contura 420
11=CompaqLTE Elite, LTE Elite 4/40C, LTE Elite 4/75CX, LTE Elite 4/75CXL
12=Compuadd425TX
13=DELLatitude
14=DELLatitude XP
```

15=IBMThinkPad 230
16=IBMThinkPad 340/CSE
17=IBMThinkPad 350
18=IBMThinkPad 355/C/CS 360/C/CE/CS/CSE/P/PE
19=IBMThinkPad 365X/XD/E/ED/C/CD/CS/CSD
20=IBMThinkPad 370C
21=IBMThinkPad 500
22=IBMThinkPad 510
23=IBMThinkPad 530CS
24=IBMThinkPad 560
25=IBMThinkPad 701C
26=IBMThinkPad 710T
27=IBMThinkPad 720/C
28=IBMThinkPad 730T
29=IBMThinkPad 750/C/CS
30=IBMThinkPad 750CE
31=IBMThinkPad 755C/CS
32=IBMThinkPad 755CD/CDV
33=IBMThinkPad 755CE/CSE/CV/CX
34=IBMThinkPad 760C/CD
35=IBMThinkPad 760E
36=IBMThinkPad 760ED
37=IBMThinkPad 760EL/ELD
38=IBMThinkPad 760X/XD
39=IBMPS/2 E
40=LexbookGS40C
41=Matsushita
42=NCRSafari
43=NECVersa
44=Panasonic
45=ToshibaT1960CS/CT
46=ToshibaT200CD\80 PEN
47=ToshibaT2100CT
48=ToshibaT2130CS/CT
49=ToshibaT2400CS/CT
50=ToshibaT3400CT
51=ToshibaT3600/T3600CT/200 Portege
52=ToshibaT4500
53=ToshibaT4600
54=ToshibaT4700
55=ToshibaT4800
56=ToshibaT4850CT
57=ToshibaT4900CT
58=ToshibaT610CT Portege
59=ToshibaT6600C
60=Zeos
61=ZeosMeridian

62=ZenithZ-lite 425L
63=INTELPIC
64=DATABOOKTCIC

- **PCMCIAOptions**

0=Don't install (DEFAULT)
1=Install all
2=Modem/FAX services
3=Hard disk services
4=FLASH services

- **PrimaryCodePage**

Specifies whether the "national" or "multilingual" code page is primary (first active code page before switching).

For many countries, there is no "real national" code page; refer to the table under keyword CountryCode on page 188. These countries usually use the US national code page (437) if PrimaryCodePage is set to 1. For most countries, it would make more sense to use the multilingual code page (usually 850) as a company default since it covers at least the European National Language Support characters to a much higher degree.

In any environment, it is a good idea to decide on a 'company' default since the users usually will share the same documents and work with the same databases.

1 = National (DEFAULT)
2 = Multilingual

- **PrinterPort**

Specifies to which printer port the default printer should be attached.

1 = LPT1 (DEFAULT)
2 = LPT2
3 = LPT3
4 = COM1
5 = COM2
6 = COM3
7 = COM4

- **ProcessEnvironment**

Specifies whether or not to add keyword/keyvalues to the environment. This makes it easy for user programs, batch files, and so forth (UserExit) to access response file data. Since we're already processing the file, they will only have to read an environment variable.

0 = Do not add keyword/keyvalues to environment
1 = Add keyword/keyvalues to environment (DEFAULT)

- **ProgressIndication**

Specifies whether or not to display a progress indicator during the installation.

0 = No progress indication
1 = Progress indication (DEFAULT)

- **RebootRequired**

Specifies if the machine should be automatically warm-booted when installation is complete. This is ignored if the `ExtendedInstall` response is specified.

0 = Ask user to reboot (DEFAULT)
1 = Auto-reboot

Note

For CID installations, it is required that `RebootRequired` is set to 0. When doing CID installations, the user is not asked to reboot; instead, the reboot is handled by the software distribution manager.

- **SCSI**

Specifies which, if any, SCSI adapter device you wish to install support for.

Valid values are:

0 = None
1 = Autodetect
2=Adaptec1460,1510,152X,AIC6260,6360
3=Adaptec154X,174X FAST SCSI
4=Adaptec1640 MCA SCSI
5=Adaptec174X EISA SCSI
6=Adaptec274X,284X,AIC 7770 FAST SCSI
7=Adaptec294X,394X, AIC 7870 PCI SCSI
8=AdaptecAPA 348, Trantor T348 MiniSCSI Parallel
9=AdaptecAPA 358, Trantor T358 MiniSCSI EPP Parallel
10=BusLogicBusMaster SCSI Adapters
11=BusLogicFlashpoint LT, LW, DL
12=DPTPM2XXX,PM32XX SmartCache,SmartRAID
13=FutureDomain TMC8XXM,TMC950
14=FutureDomain TMC3260,MCS600,700,TMC-16XX,17XX,1800
15=FutureDomain 7000EX EISA SCSI
16=IBMPC ServerRAID Adapter
17=IBMPS/2 SCSI Adapter
18=IBMSCSI II RAID Adapter
19=IBM16-Bit AT Fast SCSI Adapter
20=MylexDAC 960 RAID Adapter
21=ProAudioSpectrum 16 / Trantor SCSI

22=QLogicQLA400 Adapter
23=QLogicQLA510 Adapter
24=QLogicQLA1000 Adapter
25=TrantorT128/T228 SCSI Adapter
26=TrantorT130B 8 Bit SCSI Adapter
27=TrantorT160/T260 16 Bit SCSI Adapter
28=TrantorT338 MiniSCSI Parallel Port Adapter

- **SeedConfigSysLine**

Specifies a text line to be appended to the CONFIG.SYS written to the seed system from which PM Install boots. This will allow device drivers (that may be required) to become part of that seed system. There may be multiple occurrences of this keyword. No validity checking is done.

The key value is a valid CONFIG.SYS statement.

- **SerialDeviceSupport**

Specifies whether or not to install the device driver.

0 = Don't install
1 = Install (DEFAULT)

- **ShareDesktopConfigFiles**

Specifies that the desktop configuration files should be shared between an existing Windows system and the WIN-OS/2 system being installed. If this option is selected, the Windows desktop will be updated when changes are made to the WIN-OS/2 desktop. This option is valid only when option 1 is selected for the `WIN-OS/2Desktop` key value.

0 = Do not share the Windows desktop configuration files
1= Share the Windows desktop configuration files

- **SourcePath**

Specifies the path that should be used as a source drive and directory from which to install the disk images. This keyword is optional because you could set the `SourcePath` parameter in the CONFIG.SYS file on the LAN transport diskette. If, however, this keyword is used in the response file for the client, it will override the `SourcePath` statement in the CONFIG.SYS file on the LAN transport diskette.

The key value is drive and optional path.

Default is A:\

- **TargetDrive**

Specifies the target drive to which OS/2 should be installed. This drive is assumed to be a valid partition. If a partition other than C: is specified, it is

assumed that Boot Manager is already installed to enable booting an operating system from different partitions.

The key value is a valid partition. The default is the first acceptable partition.

- **ToolsAndGames**

Specifies which tools or games can be installed.

Valid values are:

0=Install none
1=Install all (DEFAULT)
2=Enhanced Editor
3=Search and Scan Tool
4=Optional Bit Maps
5=Solitaire - Klondike
6=Pulse
7=Chess
8=Mahjongg Solitaire

Example:

```
ToolsAndGames=2,5,6
```

This should install Enhanced editor, solitaire and Pulse.

- **UserExit**

Specifies the name of a program that can be run at the end of the install procedure. Install waits for the program to return. This keyword may occur more than once. Each will be executed in the order that they appear at the end of OS/2 Install.

The fully qualified path is required when specifying a user exit program.

The key value is an user exit program name (DEFAULT=none).

KEYVALUE=User defined version string

For an example on the use of `EarlyUserExit`, refer to "The User Exit Keywords of the Response File" on page 214.

- **WindowsInstallSourcePath**

Specifies the path to Windows diskettes in a CID directory tree.

The key value is a string that specifies the path relative to the source path where the Windows product diskettes reside on the CID tree.

Example:

```
WindowsInstallSourcePath=\
```

In the previous example, if the `SourcePath` keyword is set to `X:\IMG\OSWARP4`, this would cause the install program to look for the Windows Diskette 1 in the `X:\IMG\OS2WARP4\DISK_W1` directory.

- **WIN-OS/2Desktop**

Specifies what the WIN-OS/2 desktop should look like. Option 1 should be selected only if Windows currently exists (See related options `ExistingWindowsPath` and `WindowsInstallSourcePath`). Option 2 should be selected only if WIN-OS/2 has previously been installed.

```
0=Install standard WIN-OS/2 desktop (DEFAULT)
1=Copy existing Windows desktop and use as the WIN-OS/2 desktop
2=Preserve WIN-OS/2 desktop currently installed
```

- **WIN-OS/2Support**

Specifies whether or not to install WIN-OS/2 environment. If yes, select WIN-OS/2 groups or other components.

```
0 = Do NOT install WIN-OS/2
1 = All available groups and components (DEFAULT)
2 = WIN-OS/2 Readme File
3 = WIN-OS/2 Accessories Group
4 = WIN-OS/2 Screen Save Utility
5 = WIN-OS/2 Sound Utility
6 = WIN-OS/2 Main and StartUp Group ONLY (Minimum support)
```

Note: WIN-OS/2 Main Group and StartUp Group will be installed (mandatory) when WIN-OS/2 is supported (case 1,2,3,4,5).

Example:

```
WIN-OS/2Support=3,4
```

This would install WIN-OS/2 Main Group, StartUp Group and WIN-OS/2 Accessories and Screen Save Utility.

- **WIN-OS/2TargetDrive**

Specifies on which valid partition drive to install WIN-OS/2.

The key value is any valid FORMATTED partition. The default is C:.

Example:

```
WIN-OS/2TargetDrive=D:
```

This would install WIN-OS/2 to partition D: located in `\OS2\MDOS\WINOS2`.

7.1.3 New Keywords in OS/2 Warp 4

These keywords are valid keywords, but are not used for the installation of OS/2 through `SEINST/RSPINST`. They are related to applications loaded by the command line interface of Feature Installer (`CLIFT`).

- **ART.Selection**

Specifies whether or not to install the application registration (also known as the dancing elephant) option.

0 = Don't install
1 = Install (DEFAULT)

- **BASKPSP** (BonusPak Ask PSP Knowledge Database)

Specifies whether or not to install BonusPak AskPSP support during the installation. In total, there are two keywords to be specified:

- **BASKPSP.Selection**

0 = do NOT install AskPSP support
1 = install AskPSP support (DEFAULT)

- **BASKPSP.TarDrv**

Specifies the target drive in which AskPSP is to be installed. The key value is a valid drive letter.

C:

- **BPCIM** (BonusPak Compuserve Information Manager)

Specifies whether or not to install Compuserve files during the installation. In total, there are two keywords to be specified:

- **BPCIM.Selection**

0 = do NOT install Compuserve files
1 = install Compuserve files (DEFAULT)

- **BPCIM.TarDrv**

Specifies the target drive in which Compuserve is to be installed. The key value is a valid drive letter.

C:

- **BPFAXWORKS**

Specifies whether or not to install FaxWorks files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPFAXWORKS.Selection**

0 = do NOT install FaxWorks files

1 = install FaxWorks files (DEFAULT)

- **BPFAXWORKS.TarDrv**

The key value is a valid drive letter.

- **BPHAL**

Specifies whether or not to install HyperAccess Lite files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPHAL.Selection**

0 = do NOT install HyperAccess Lite files

1 = install HyperAccess Lite files (DEFAULT)

- **BPHAL.TarDrv**

The key value is a valid drive letter.

- **BPHPJETCLIENT**

Specifies whether or not to install Jet Admin Client files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPHPJETCLIENT.Selection**

0 = do NOT install Jet Admin Client files (DEFAULT)

1 = install Jet Admin Client files

- **BPHPJETCLIENT.TarDrv**

The key value is a valid drive letter.

- **BPHPJETSERVER**

Specifies whether or not to install Jet Admin Server files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPHPJETSERVER.Selection**

0 = do NOT install Jet Admin Server files (DEFAULT)

1 = install Jet Admin Server files

- **BPHPJETSERVER.TarDrv**

The key value is a valid drive letter.

- **BPIBWORKS**

Specifies whether or not to install IBM Works files during the installation. It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPIBWORKS.Selection**

- 0 = do NOT install IBM Works files
 - 1 = install IBM Works files (DEFAULT)

- **BPIBWORKS.TarDrv**

- The key value is a valid drive letter.

- **BPMARKNET**

Specifies whether or not to install MarkNet Port Driver files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPMARKNET.Selection**

- 0 = do NOT install MarkNet Port Driver files (DEFAULT)
 - 1 = install MarkNet Port Driver files

- **BPMARKNET.TarDrv**

- The key value is a valid drive letter.

- **BPMARKVIS**

Specifies whether or not to install MarkVision files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPMARKVIS.Selection**

- 0 = do NOT install MarkVision files (DEFAULT)
 - 1 = install MarkVision files

- **BPMARKVIS.TarDrv**

- The key value is a valid drive letter.

- **BPRS2**

Specifies whether or not to install Remote Support files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPRS2.Selection**

- 0 = do NOT install Remote Support files

1 = install Remote Support files (DEFAULT)

- **BPRS2.TarDrv**

The key value is a valid drive letter.

- **BPVIDEOIN**

Specifies whether or not to install VideoIn files during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **BPVIDEOIN.Selection**

0 = do NOT install VideoIn files

1 = install VideoIn files (DEFAULT)

- **BPVIDEOIN.TarDrv**

The key value is a valid drive letter.

- **COACHES.Selection**

Specifies whether or not to install the Warp guide option.

0 = Don't install

1 = Install (DEFAULT)

- **DAXCOMP1**

Specifies whether or not to install the Developer API Extensions during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **DAXCOMP1.Selection**

0 = do NOT install

1 = install (DEFAULT)

- **DAXCOMP1.TarDrv**

The key value is a valid drive letter.

- **HOTPLUG.Selection**

Specifies whether or not to install Support for an External Floppy Drive for a laptop.

0=Don't install (DEFAULT)

1=Install

- **JAVASMPLS.Selection**

Specifies whether or not to install the JAVA sample files.

0 = Don't install

1 = Install (DEFAULT)

- **JAVATLKT.Selection**

Specifies whether or not to install the JAVA toolkit option.

0 = Don't install

1 = Install (DEFAULT)

- **ODBASE**

Specifies whether or not to install OpenDoc during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **ODBASE.Selection**

0 = do NOT install

1 = install (DEFAULT)

- **ODBASE.TarDrv**

The key value is a valid drive letter.

- **ODSECBASE**

Specifies whether or not to install the Security Extensions Services during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **ODSECBASE.Selection**

0 = do NOT install

1 = install (DEFAULT)

- **ODSECBASE.TarDrv**

The key value is a valid drive letter.

- **SRVDIAG.Selection**

Specifies whether or not to install the serviceability and diagnostic aids option.

0 = Don't install

1 = Install (DEFAULT)

- **SRVDOC.Selection**

Specifies whether or not to install the serviceability documentation option.

0 = Don't install

1 = Install (DEFAULT)

- **SYSMGT.Selection**

Specifies whether or not to install the system management option.

0 = Don't install
1 = Install (DEFAULT)

- **VT**

Specifies whether or not to install the Voice Type option during the installation.

It has two subkeywords to indicate if it is or is not to install. The second keyword provides a valid drive where to install:

- **VT.Selection**

- 0 = do NOT install
1 = install (DEFAULT)

- **VT.TarDrv**

- The key value is a valid drive letter.

- **WARMDOCK.Selection**

Specifies whether or not to install Support for Dock II docking station for an IBM Thinkpad.

0=Don't install (DEFAULT)
1=Install

- **WARMSWAP.Selection**

Specifies whether or not to install Support for floppy and CD-ROM Swapping for UltraBay devices on an IBM Thinkpad.

0=Don't install (DEFAULT)
1=Install

Keyword WARMPUG.Selection Not Supported!

The SAMPLE.RSP file documents a keyword that actually is not supported:

WARMPUG.Selection

This keyword has been replaced by:

WARMSWAP.Selection

7.1.4 Printer Description Table PRDESC.LST

A list of printers (PRDESC.LST) resides on the first printer device driver Diskette 1 (PMDD_1). On an installed system, it can be found in the \OS2\INSTALL directory. This list is to be used for the `AdditionalPrinters` and

DefaultPrinter keywords. Please remember that this list is version dependent since more printer drivers are added with each upgrade. The DefaultPrinter key value is used as an index in the list.

DefaultPrinter Example

DefaultPrinter = 3 means that the installed printer driver will be PSCRIPT.DRV for the Apple LaserWriter.

Note: The list below is not complete and does not necessarily reflect the current list on the first printer device driver diskette of your OS/2 country NLS version.

```
AST TurboLaser: AST TurboLaser (PSCRIPT.DRV)
AAgfa/Compugraphic 400PS: Agfa/Compugraphic 400PS (PSCRIPT.DRV)
Apple LaserWriter: Apple LaserWriter (PSCRIPT.DRV)
Apple LaserWriter 16/600 PS: Apple LaserWriter 16/600 PS (PSCRIPT.DRV)
Apple LaserWriter II NT: Apple LaserWriter II NT (PSCRIPT.DRV)
Apple LaserWriter II NTX: Apple LaserWriter II NTX (PSCRIPT.DRV)
Apple LaserWriter Plus: Apple LaserWriter Plus (PSCRIPT.DRV)
Brother HJ-100i 24 pin - 80 columns: Brother HJ-100i (OMNI.DRV)
Brother HJ-400: Brother HJ-400 (OMNI.DRV)
...
...
...
Tektronix Phaser PX: Tektronix Phaser PX (PSCRIPT.DRV)
Tektronix Phaser PXi: Tektronix Phaser PXi (PSCRIPT.DRV)
Varityper VT-600: Varityper VT-600 (PSCRIPT.DRV)
Wang LCS15: Wang LCS15 (PSCRIPT.DRV)
Wang LCS15 FontPlus: Wang LCS15 FontPlus (PSCRIPT.DRV)
XEROX 4215/MRP V1: XEROX 4215/MRP V1 (PSCRIPT.DRV)
XEROX 4219/MRP V1: XEROX 4219/MRP V1 (PSCRIPT.DRV)
Xerox 4003/4004: Xerox 4003/4004 (OMNI.DRV)
```

Figure 51. Printer Description List PRDESC.LST

In total, there are 641 printer driver entries.

Updated OS/2 Printer Drivers

To obtain the latest OS/2 printer drivers from the Internet, point your Web browser to the following Web site:

<ftp://ftp.software.ibm.com/ps/products/os2/fixes/printerpak>

7.1.5 Include Keywords Processing Sequence

The following part explains Include keywords in detail and describes how individual response files and results can be managed. The keywords described are `Include` and `IncludeInLine`.

7.1.5.1 Single Include Example

The response file provides a keyword called `Include`. `Include` makes it possible to specify another response file which can be used to include additional keywords or override the current response file's keywords. Different include files could therefore be used for those users who have specific requirements not provided by the standard response file.

The example below provides a sample response file `STANDARD.RSP` that uses the `include` keyword to include a response file called `INDIV.RSP`.

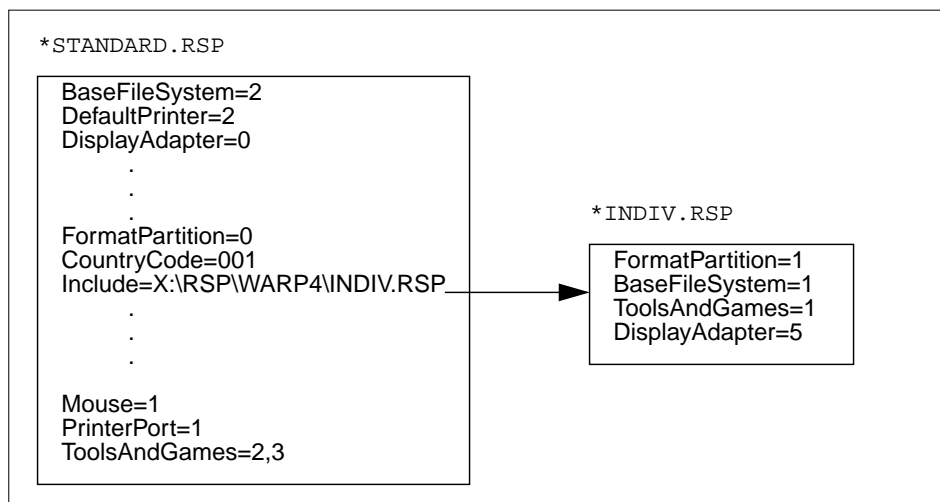


Figure 52. Include Example

Note

The entire `STANDARD.RSP` response file will be processed from beginning to end, and only then will the include file, namely `INDIV.RSP`, be interpreted.

In the example above, the workstation using the `INDIV.RSP` include file will differ in the sense that its hard disk will be formatted using the HPFS file system. It will use a VGA adapter instead of accepting the default and will have all tools and games installed on it.

7.1.5.2 Single IncludeInLine File Example

The response file provides a keyword called `IncludeInLine`. `IncludeInLine` makes it possible to specify another response file which can be used to include additional keywords or override the current response file's keywords. The example below provides the same sample response file `STANDARD.RSP`, that now uses the `IncludeInLine` keyword to include a response file called `INDIV.RSP`.

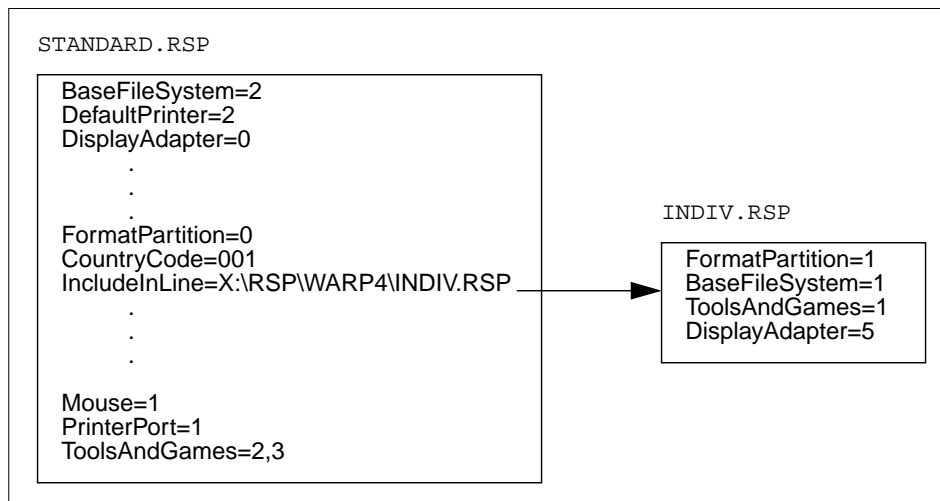


Figure 53. `IncludeInLine` Example

Note

The `STANDARD.RSP` response file will be processed from the beginning until it reaches the `IncludeInLine` keyword. The include file, `INDIV.RSP`, will then be interpreted from beginning to end. `STANDARD.RSP` processing will resume after `INDIV.RSP` has been interpreted.

In the example above, the workstation using the `INDIV.RSP` include file will differ in the sense that its hard disk will be formatted using the HPFS file system. It will use a VGA adapter instead of accepting the default, but only the tools corresponding to `ToolsAndGames=2,3` will be installed because `ToolsAndGames=2,3` found in `STANDARD.RSP` overrides `ToolsAndGames=1` specified in `INDIV.RSP`.

7.1.5.3 Multiple Include Files Example

The following example shows the functioning of multiple include statements within the response file. This will be explained by discussing the sample.

The sample consists of five include files called MOUSE1.INC to MOUSE5.INC. Each one has one `ConfigSysLine` with a remark telling which MOUSEx.INC was called and the appropriate mouse function number. All the files can contain other keywords that will be interpreted in the same way.

The content of file MOUSEx.INC is shown below:

```
ConfigSysLine=REM mousex include passed
mouse=x
```

Where `x` is 1, 3, 4 or 5.

The MOUSE2.INC has one `ConfigSysLine` with a remark plus two `Include` statements. The content of file MOUSE2.INC is displayed below:

```
ConfigSysLine=REM mouse2 include passed
include=mouse4.inc
include=mouse5.inc
```

The sample response file, INCSAMP.RSP, is shown below:

```
include=MOUSE1.INC
include=MOUSE2.INC
include=MOUSE3.INC
mouse=0
```

After all data within the basic response file has been interpreted (including the `mouse=0` statement in the above sample), the program continues by reading the include files in the order of their appearance. (MOUSE1.INC, MOUSE2.INC, MOUSE3.INC, and so forth). Inside the MOUSE2.INC include files, it finds more `Include` statements, which it stores at the end of a program internal include table. After MOUSE3.INC has been processed, processing will continue by reading the MOUSE4.INC file. Eventually, MOUSE5.INC is the last include file being read.

Even though the `mouse=0` statement in the above INCSAMP.RSP file came after all include statements, it will be interpreted before the include files are handled.

The following listing shows the essential part of INSTALL.LOG:

```
Processing: include=MOUSE1.INC
Processing: include=MOUSE2.INC
Processing: include=MOUSE3.INC
Processing: mouse=0
```

```

Processing: ConfigSysLine=REM mouse1 include passed
Processing: mouse=1
Processing: ConfigSysLine=REM mouse2 include passed
Processing: include=mouse4.inc
Processing: include=mouse5.inc
Processing: ConfigSysLine=REM mouse3 include passed
Processing: mouse=3
Processing: ConfigSysLine=REM mouse4 include passed
Processing: mouse=4
Processing: ConfigSysLine=REM mouse5 include passed
Processing: mouse=5

```

The following figure visualizes the process explained above.

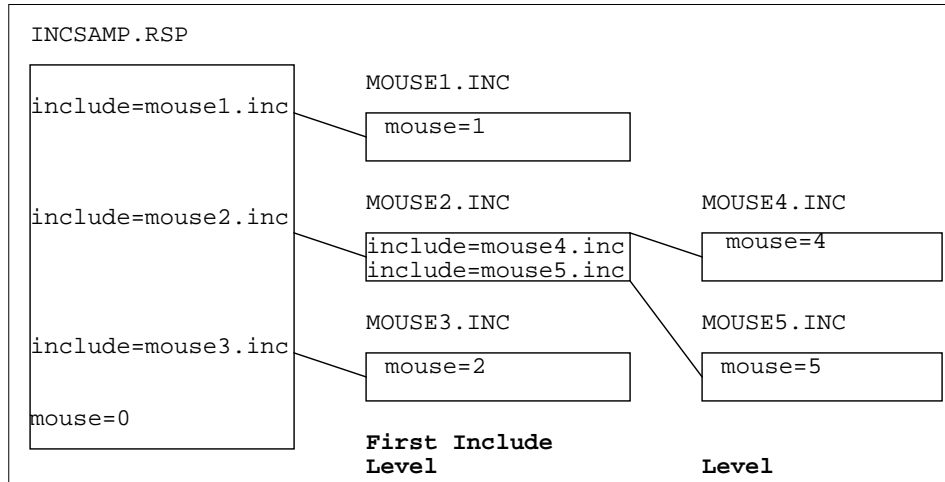


Figure 54. Embedded Include Keywords Example

7.1.6 The User Exit Keywords of the Response File

The response file has two exit keywords:

- EarlyUserExit
- UserExit

The `EarlyUserExit` is called at the very beginning before any action is taken upon the keywords in the response file.

The `UserExit` is executed at the very end, just before the reboot keyword is processed.

These user exits can be used to run any kind of program or CMD file as long as it is in the path of the `PATH` statement in the `CONFIG.SYS`.

The following section uses an example to describe the use of both exits.

If the target drive is going to be formatted and some files which need to be saved were brought into the system after the previous install, they need to be saved before the formatting takes place and be restored at the end of the installation process.

For example, assume the IBM 4717 MSRE support for the magnetic stripe reader was added later and is also needed in the new installation. These files need to be backed up and restored at the end of the installation.

If a normal command procedure called `USERSAFE.CMD` was used, the response file statement used should read:

```
EarlyUserExit=USERSAFE.CMD
```

This command procedure could be used by every workstation if it resides on the server.

The following example shows how MSRE-code can be saved and reinstalled.

The first example shows the actual copying during the `EarlyUserExit` CMD execution.

```
IF EXIST C:\OS2\FIOAUXDD.SYS      COPY C:\OS2\FIOAUXDD.SYS D:\OS2SAV\*. *
IF EXIST C:\OS2\DLL\MAGCALLS.DLL COPY C:\OS2\DLL\MAGCALLS.DLL D:\OS2SAV\DLL\*. *
IF EXIST C:\OS2\SYSTEM\FIO.MSG    COPY C:\OS2\SYSTEM\FIO.MSG D:\OS2SAV\MSG\*. *
```

Instead of `D:\OS2SAV`, a redirected drive on the server could be used. This drive has to be a **per client** drive to ensure that each client has a unique directory to copy to and restore its files from. Otherwise multiple clients could access the same redirected drive, thereby disrupting the process of copying because not just the files unique to one client workstation would be in this directory.

At the end of the installation process, the second exit is called which then starts a second command procedure, REXX procedure or program. Assuming the command procedure is named `USERREST.CMD`, the syntax for this response file statement would be:

```
UserExit=USERREST.CMD
```

Following the above rules, the content of this file should be:

```
IF EXIST D:\OS2SAV\FIOAUXDD.SYS    COPY D:\OS2SAV\FIOAUXDD.SYS C:\OS2\*. *
IF EXIST D:\OS2SAV\DLL\MAGCALLS.DLL COPY D:\OS2SAV\DLL\MAGCALLS.DLL C:\OS2\DLL\*. *
```

```
IF EXIST D:\OS2SAV\SYSTEM\FIO.MSG COPY D:\OS2SAV\SYSTEM\FIO.MSG C:\OS2\SYSTEM\*.*
```

By the command `IF EXIST`, it makes sure that the copy to/from `\OS2SAV` only takes place when a specific file exists in that specific directory.

Note: All files on the target drive for the "saves" in the `USERSAFE.COMD` should be deleted after completion of the copy statement by issuing the following command:

```
ECHO Y DEL V:\*.*
```

assuming the `v:` drive is being the redirected drive where the files reside. The `ECHO Y` does the rerouting of the "Yes" on the question asked by the `DEL` command when a `del *.*` is issued.

If every system on one logical LAN has to be equipped with the same files, which are not distributed with OS/2, these files can be placed on the server in a subdirectory of the distribution image directory (redirected `Z:` - drive) and copied during `UserExit` execution.

Assuming the `MSRE` service is needed on every workstation in one physical LAN and will reside in a subdirectory called `\CUSTDSK`, the above file (`USERREST.COMD`) could be used as follows:

```
COPY Z:\CUSTDSK\FIOAUXDD.SYS C:\OS2\*.*
COPY Z:\CUSTDSK\MAGCALLS.DLL C:\OS2\DLL\*.*
COPY Z:\CUSTDSK\FIO.MSG C:\OS2\MSG\*.*
```

Note: By doing this, the installation of clients and servers can be extended to add new program packages or utilities during installation of the OS/2 base operating system.

7.2 Feature Installer Response File

The Feature Installer (FI) uses two response files in total:

- `\OS2\INSTALL\FIBASE.RSP`
- Original OS/2 base operating system used by `SEINST/RSPINST`; This response file is called a partial response file by the Feature Installer.

The `FIBASE.RSP` file is a more than 1 MB long ASCII file. It must not be changed. To customize the products' installation, it is necessary to specify the so-called partial response file that specifies BonusPak components and other OS/2 Warp 4 features to be installed.

Figure 55 shows an extract of `FIBASE.RSP` response file:


```

ART=(
    Selection=0
)
BPFAXWORKS=(
    Selection=0
)
BPIBMWORKS=(
    Selection=1
    Variable=(
        Name=TarDrv
        Value=d:
    )
)

```

Figure 55. Feature Installer Base Response File FIBASE.RSP (Extract)

The default option for installing the BonusPak products in FIBASE.RSP is documented in 7.1.3, “New Keywords in OS/2 Warp 4” on page 204. To install a product, give a value 1 to `Selection`. To change the default installation disk, give the drive unit to `value`.

In the preceding example, products will be installed according to their default values, but ART and FAXWORKS will not be installed, and IBMWORKS will be installed on drive D:.

7.2.1 Example of an OS/2 Warp 4 Response File

Following is a sample response file that was used in our scenarios. First, the response file name was passed as an parameter to SEINST.EXE. Right after SEINST.EXE, the Feature Installer (CLIFI.EXE) will use the very same response file along with a base Feature Installer response file, FIBASE.RSP.

It is very important to understand that Feature Installer (CLIFI.EXE) is executed automatically by SEINST.EXE invoked through LAN CID Utility. However, when using NetView DM/2 or TME 10 SD 3.1.3, you need to run Feature Installer separately as another software package. Since Feature Installer needs PM (Presentation Manager) to be active, CLIFI.EXE can only be executed when the remote workstation reboots with MPTS and NVDM/2 or SD 3.1.3 client active. Find more details about Feature Installer in “ASD - Alternative Software Delivery” on page 154.

```

FormatPartition=1
BaseFileSystem=1
SerialDeviceSupport=1
DisplayAdapter=0
AlternateAdapter=0
CDROM=2
DefaultPrinter=396
APM=0
Optical=0
Infrared=0
PCMCIA=0
Documentation=1
Fonts=1
OptionalSystemUtilities=1
OptionalSystemComponents=1
ToolsAndGames=2,3,4,6
DOSSupport=1
WIN-OS/2Support=1
DPMI=1
ExitOnError=1
MultimediaSupport=0
RebootRequired=1
PrimaryCodePage=1
ProgressIndication=1
PrinterPort=1
MousePort=0
MigrateConfigFiles=1
CountryCode=001
CountryKeyboard=US
ConfigSysLine=SET RESTARTOBJECTS=STARTUPFOLDERONLY
ConfigSysLine=PAUSEONERROR=NO
ConfigSysLine=AUTOFAIL=YES
* CLIFI-related keywords for Phase 2 of OS/2 Warp 4 Install
HOTPLUG.Selection=0
WARMDOCK.Selection=0
WARMSWAP.Selection=0
* Note: The originally documented keyword WARMPLUG in SAMPLE.RSP
*       is NOT supported. Use WARMSWAP instead!
COACHES.Selection=1
ODBASE.Selection=1
ODBASE.TarDrv=C:

```

Figure 56. Example of an OS/2 Response File (Part 1 of 2)

```

VT.Selection=0
VT.TarDrv=C:
ODSECBASE.Selection=0
ODSECBASE.TarDrv=C:
ART.Selection=0
JAVARUN.Selection=1
JAVASMP.LS.Selection=1
JAVATLKT.Selection=1
* BonusPak Selection Installation through CLIFT
BPCIM.Selection=0
BPCIM.TarDrv=C:
BPHAL.Selection=0
BPHAL.TarDrv=C:
BPIBMWORKS.Selection=0
BPIBMWORKS.TarDrv=C:
BPFAXWORKS.Selection=0
BPFAXWORKS.TarDrv=C:
BPVIDEOIN.Selection=0
BPVIDEOIN.TarDrv=C:
BPASKPSP.Selection=0
BPASKPSP.TarDrv=C:
BPRS2.Selection=0
BPRS2.TarDrv=C:
BPHPJETCLIENT.Selection=0
BPHPJETCLIENT.TarDrv=C:
BPHPJETSERVER.Selection=0
BPHPJETSERVER.TarDrv=C:
BPMARKVIS.Selection=0
BPMARKVIS.TarDrv=C:
BPMARKNET.Selection=0
BPMARKNET.TarDrv=C:

```

Figure 57. Example of an OS/2 Response File (Part 2 of 2)

7.3 MPTS Response File

MPTS provides a utility called `LAPSRSP` to build response files for unattended installation of MPTS.

`LAPSRSP.EXE` needs MPTS configuration files, for example the `PROTOCOL.INI` file, as input.

There are two basic approaches providing these configuration files:

1. Use the configuration files from a running system.

2. Build new configuration files using the MPTS installation program.

LAPSRSP.EXE does not verify that the configuration files are valid. If you provide invalid configuration files, you will get invalid response files that need further editing!

Place MPTS response files in directory \CID\RSP\MPTS.

7.3.1 Building a Configuration File

Following section introduces two ways of creating configuration files that are used to create response files.

7.3.1.1 Configuration Files from a Running System

You can use a PROTOCOL.INI from a different running system that has the same kind of network adapter. If you do that, please remember to change the network adapter address (if applicable).

Valid MPTCONFIG.INI, RFCNAMES.LST, RFCBCST.LST, and RESOLV files from running systems can also be input to LAPSRSP.

7.3.1.2 Configuration Files Using MPTS Installation Program

Follow the instructions below:

1. Use a code server or on any other workstation with the same version of MPTS.

Note

The type of LAN adapter you wish to do a PROTOCOL.INI file for does not need to be installed in this workstation.

2. Make a backup copy of your current MPTS configuration files because the following steps will generate a new files.

To be on the safe side, make a backup copy of the active CONFIG.SYS also. Assuming that C is the boot drive and that MPTS is installed on C, execute:

```
COPY C:\config.sys *.bak
COPY C:\ibmcom\protocol.ini *.bak
COPY C:\ibmcom\rfc*.lst *.bak
COPY C:\mptn\bin\mptconfig.ini *.bak
COPY C:\mptn\bin\mptstart.cmd *.bak
COPY C:\mptn\bin\startup.cmd *.bak
COPY C:\mptn\etc\resolv *.bak
```

Depending on your current MPTS configuration, it is possible that you do not have all the files mentioned above.

3. Invoke MPTS.
4. Select **Configure**.
5. Ensure that radio button **LAN adapters and protocols** is selected below Adapter and Protocols. Click on **Configure**.
6. Remove current protocols and network adapters from the current configuration. Note that you have to remove all protocols first, before attempting to remove the adapter.
7. Select the LAN adapter you want and add it to the current configuration.
8. Select the **IBM OS/2 NetBIOS** protocol and add it to the current configuration.
9. Select **IBM IEEE 802.2** protocol and add it to current configuration if you intend to install any application running on top of the IEEE 802.2 interface. If the PROTOCOL.INI will be used as input for THINLAPS later, the IEEE 802.2 protocol should not be added.
10. Select the LAN adapter from the current configuration and click on **Edit**.
11. Fill in the values, at least for Shared RAM Address if the PROTOCOL.INI is intended for ISA-bus clients.

Make sure that these values do not conflict with any other values used for devices attached to the system for which you are creating this PROTOCOL.INI.

12. Fill in the Network Adapter Address if you want to use locally administered addresses (LAA).
13. Fill in other values you wish to set for the adapter and the protocols.

If the client will be a NetView DM/2 V2.1 client, you may want to check the values for some of the NetBIOS protocol resources. The additional requirements for a NetView DM/2 V2.1 client are two sessions, 14 commands and six names.

14. Click on **OK**.
15. If all that was needed was a new PROTOCOL.INI for a different adapter, nothing more has to be done in this panel. Select **Close** to leave the Configure panel.

If you need any other configuration steps (for TCP/IP for example), insert them here. When finished, click on **Close** to leave the Configure panel.

16. Click on **Exit** to leave the Multi-Protocol Transport Services panel.
17. Make sure that Update CONFIG.SYS is **not** checked and select **Exit**.
18. Save the PROTOCOL.INI created under another name:

```
COPY C:\IBMCOM\PROTOCOL.INI \IBMCOM\PROTOCOL.MOD
```

and for MPTS, save the other configuration files as well (not all of them are necessarily created; it depends on the configuration).

```
COPY C:\ibmcom\rfc*.lst *.mod
COPY C:\mptn\bin\mptconfig.ini *.mod
COPY C:\mptn\bin\mptstart.cmd *.mod
COPY C:\mptn\bin\startup.cmd *.mod
COPY C:\mptn\etc\resolv *.mod
```

19. Restore the original versions of CONFIG.SYS and configuration files:

```
COPY C:\config.bak *.sys
COPY C:\ibmcom\protocol.bak *.ini
COPY C:\ibmcom\rfc*.bak *.lst
COPY C:\mptn\bin\mptconfig.bak *.ini
COPY C:\mptn\bin\mptstart.bak *.cmd
COPY C:\mptn\bin\startup.bak *.cmd
COPY C:\mptn\etc\resolv.bak *
```

Now you are ready to use `LAPSRSP`, which can be found in the `\IBMCOM` subdirectory of an installed MPTS or in the `\CID\IMG\MPTS` directory.

LAPSRSP Syntax

```
LAPSRSP <source path> <target path> <options>
```

<i><source path></i>	Specifies drive and path of source PROTOCOL.INI file.
<i><target path></i>	Specifies drive and path for the resulting LAPS response file.
<i><options></i>	The following parameters are optional parameters:
<i>/T:</i>	optional Value of the MPTS response file Target parameter. This is the OS/2 drive.
<i>/I:</i>	optional Value of the MPTS response file <code>Install</code> keyword. This has two values, <code>PRODUCT</code> or <code>ADDITIONAL</code> .

Normally `PRODUCT` is used. `ADDITIONAL` should only be used when adding something to an existing MPTS installation.

`/C:` optional

Value of the MPTS response file `CFG_PATH_NAME` parameter. This is the OS/2 fully qualified file name of the OS/2 EE V1.3 Communications Manager or CM/2 .CFG file that will be migrated to a `PROTOCOL.INI` file.

`/U:` optional

Value of the MPTS response file `UPGRADE_LEVEL` keyword. This has three values `OLD`, `SAME`, or `NEW`. When the MPTS installation is run, this keyword determines whether MPTS installs or not.

`NEW` MPTS installs regardless of existing version on the workstation.

`OLD` MPTS installs if the existing version is older than the version you are installing with.

`SAME` MPTS installs if the existing version is older than or the same as the version you are installing with.

`/M:` optional

Specify the fully qualified path and file name of a valid MPTS configuration file from a running MPTS installation (where it is found in the `\MPTN\BIN` directory as `MPTCONFIG.INI`). The `MPTCONFIG.INI` is used as input for the MPTS section of the response file.

`/N:` optional

Specify fully qualified path and file name of a valid names list file from a running MPTS installation (where it is found in the `\IBMCOM` directory as `RFCNAMES.LST`).

A valid `RFCNAMES.LST` file could look like:

```
"RJBIBM"           rjbibm.austin.ibm
"aus vols"         aus.vols.austin
"STARFLEET "      9.3.40.201
"STEINI"           9.3.40.201
"DATA \x00\x0a\xff" 9.3.40.201
```

The names list file is used within the names list section of the response file.

/B: optional

Specify the fully qualified path and file name of a valid broadcast list file from a running MPTS installation (where it is found in the \IBMCOM directory as RFCBCST.LST).

A valid RFCBCST.LST file could look like:

```
129.35.95.255
rjbibm.austin.ibm
aus.vols.austin
```

The broadcast list file is used within the broadcast list section of the response file.

/V: optional

Specify the fully qualified path and file name of a RESOLV list file from a running MPTS installation (where it is found in the \MPTN\ETC directory as RESOLV).

A valid RFCBCST.LST file could look like:

```
nameserver 9.3.1.81
```

The RESOLV file is used within the RESOLV list section of response file.

The following single-line command example assumes the PROTOCOL.MOD was created as described earlier.

LAPSRSP Example

```
LAPSRSP C:\ibmcom\protocol.mod
        D:\cid\rsp\laps\lapsrsp.rsp
        /I:product
        /T:C:
        /U:new
        /M:C:\mptn\bin\mptconfig.mod
        /N:C:\ibmcom\rfcnames.mod
        /B:C:\ibmcom\rfcbcst.mod
        /V:C:\mptn\etc\resolv.mod
```

7.3.2 Use of Client-Specific Response File for MPTS Installation

For MPTS configurations, where you want to use locally administered LAN addresses, you have to have client-specific response files.

Consider how you can group users with similar requirements and provide one default model for each group. Similar requirements are, for example, the same type of LAN adapter and the same protocols installed. There are a couple of options for MPTS on how to provide client-specific response files:

1. You can make one complete response file for each client.

This is not such a good idea if you have lots of users and at some time might want to do a global change for all of them.

2. You can use the `INCLUDE` keyword to include a default response file and then further down in the client's response file, set the keywords you want to be specific.

The INCLUDE Keyword in MPTS Response Files

When using an `INCLUDE` in an MPTS response file, the processing of the keywords in the included file is done immediately. The behavior is the same as for the `IncludeInLine` keyword for OS/2.

If a qualified path is not specified, the search order of the include files is:

- The path specified in the `/G:` parameter.
`/G` is given when MPTS is invoked.
- Current directory.
- Each path as set in the `PATH` statement on the client's `CONFIG.SYS`.
- Each path as set in the `DPATH` statement on the client's `CONFIG.SYS`.

Assuming that you have a common default response file, MPTSRSP.RSP, one client-specific file for the client ALEX to change the net address could look like:

```
; Alex MPTS response file
INCLUDE = X:\RSP\MPTSRSP.RSP
PROT_SECTION = (
nif = LANDD.NIF
section_name = LANDD_nif
NETADDRESS = "T400000001344"
```

As you can see, the MPTS response file is slightly different from those for the other products because you have to define the section and the mandatory keywords for that section in addition to the keyword(s) you really want to change.

7.3.3 Example of an MPTS Response File

The following figure displays a MPTS response file that was used in our scenarios. IEEE 802.2 (DLC), TCP/IP, and NetBEUI on logical LAN adapter 0 and TCPBEUI on logical LAN adapter 1 was configured.

```

INST_SECTION = (
  TARGET = C:
  INSTALL = PRODUCT
  UPGRADE_LEVEL = NEW
)

PROTOCOL = (
[PROT_MAN]

  DRIVERVERNAME = PROTMAN$

[IBMLXCFG]

  landd_nif = landd.nif
  netbeui_nif = netbeui.nif
  tcpbeui_nif = tcpbeui.nif
  tcpip_nif = tcpip.nif
  IBMTOK_nif = IBMTOK.NIF

[NETBIOS]

  DriverName = netbios$
  ADAPTER0 = netbeui$,0
  ADAPTER1 = tcpbeui$,1

[landd_nif]

  DriverName = LANDD$
  Bindings = IBMTOK_nif
  ETHERAND_TYPE = "I"
  SYSTEM_KEY = 0x0
  OPEN_OPTIONS = 0x2000
  TRACE = 0x0
  LINKS = 16
  MAX_SAPS = 6
  MAX_G_SAPS = 0
  USERS = 6

```

Figure 58. Example of an MPTS Response File (Part 1 of 3)

```

TI_TICK_G1 = 255
T1_TICK_G1 = 15
T2_TICK_G1 = 3
TI_TICK_G2 = 255
T1_TICK_G2 = 25
T2_TICK_G2 = 10
IPACKETS = 250
UIPACKETS = 100
MAXTRANSMITS = 6
MINTRANSMITS = 2
TCBS = 64
GDTS = 30
ELEMENTS = 1200
NETFLAGS = 0x0

[netbeui_nif]

DriverName = netbeui$
Bindings = IBMTOK_nif
ETHERAND_TYPE = "I"
USEADDRREV = "YES"
OS2TRACEMASK = 0x0
SESSIONS = 128
NCBS = 192
NAMES = 24
SELECTORS = 50
USEMAXDATAGRAM = "NO"
ADAPTRATE = 1000
WINDOWERRORS = 0
MAXDATARC = 4168
TI = 30000
T1 = 1000
T2 = 200
MAXIN = 1
MAXOUT = 1
NETBIOS_TIMEOUT = 500
NETBIOS_RETRIES = 3
NAMECACHE = 1000
RNDOPTION = 1
PIGGYBACKACKS = 1
DATAGRAMPACKETS = 12
PACKETS = 335
LOOPPACKETS = 8

```

Figure 59. Example of an MPTS Response File (Part 2 of 3)

```

PIPELINE = 5
MAXTRANSMITS = 6
MINTRANSMITS = 2
DLCRETRIES = 10
FCPRIORITY = 5
NETFLAGS = 0x0

[tcpeui_nif]

DriverName = tcpeui$
Bindings = ,IBMTOK_nif
NODETYPE = "H-Node"
NBNSADDR = "9.3.1.81"
NBDDADDR = "9.3.1.81"
OS2TRACEMASK = 0x0
SESSIONS = 128
NCBS = 192
NAMES = 24
SELECTORS = 30
USEMAXDATAGRAM = "NO"
NETBIOS_TIMEOUT = 500
NETBIOSRETRIES = 2
NAMECACHE = 1000
PRELOADCACHE = "NO"
NAMESFILE = 0
DATAGRAMPACKETS = 20
PACKETS = 100
INTERFACERATE = 300

[tcPIP_nif]

DriverName = TCPIP$
Bindings = IBMTOK_nif

[IBMTOK_nif]

DriverName = IBMTOK$
MAXTRANSMITS = 6
RECVBUFS = 2
RECVBUFSIZE = 256
XMITBUFS = 1

)

```

Figure 60. Example of an MPTS Response File (Part 3 of 3)

7.4 OS/2 Peer Response File

Like all other response files for CID enabled products, the response files for LAN Peer are ASCII files that contain sequences of keyword-value pairs. These are interpreted during the installation and configuration process.

To create response files for LAN Peer, it is recommended to use the PEERRMT program. This program resides on the code server and should be found in the \CID\IMG\OS2PEER directory if the code server has been installed by copying the OS/2 Warp 4 CD-ROM.

The following section takes the reader through the steps for generating OS/2 Peer response files.

7.4.1 Building a OS/2 Peer Response File

1. At the code server, open an OS/2 window, change to the \CID\IMG\OS2PEER directory, and enter the PEERRMT command.
2. Click on **OK** on the first screen.
3. Select the radio button for **Create a response file for a remote installation** at the Installation Tasks panel and click on **OK**.
4. At the "Response File Name" window, type a path and file name where you wish to store the response file. For example:

```
D:\CID\RSP\OS2PEER\MYRSP.RSP
```

Select **OK**.
5. Ensure that you have specified the correct path/filename and select **Yes** at the pop-up confirmation window.
6. At the "Source Drive" window, select whether you do or do not want the installation program to update existing LAN Services on the target workstation. If installing on a system without a previous copy of LAN Services, select not to update. The update option is for migrating a currently installed LAN Services to the new level being installed. Click on **OK**.
7. At the "Hard Disk" window, specify the disk letter of the target workstation where the LAN Services should be installed or updated. Select **OK**.
8. On the "LAN adapter" panel, choose the adapters on which you will run the LAN peer program. In most cases, the peer station will have only one adapter. In that case, use the default "use target setting".

9. On the "Peer Workstation Name" panel, give the name of the workstation. This name is mandatory on a pristine installation. It should be unique on the network. Select **OK**.
10. On the next panel "Peer Workstation Description", you can give information to describe the workstation. Select **OK**.
11. On the "Domain Name" panel, provide the name of the domain to which the workstation's user will have to connect. This information is mandatory for a pristine installation. Select **OK**.
12. On the LAN Server Administration graphical interface, click on **YES** if the workstation is to be used by a LAN Server administrator and this administrator uses the graphical interface. Otherwise, click on **NO**.
13. On the "NET.ACC" panel, choose **NO** if you want to keep a previous definition of users, groups and user access profiles defined for the peer access. It has no impact on the definitions on the LAN server. If you choose **YES**, all previous definitions will be destroyed. On a pristine installation, there is no previous definitions; so click on the default option.
14. On the next panel, click on **OK**, then **EXIT**, or loop to build another response file.

7.4.2 Example of an OS/2 Peer Response File

In the following example, `srvcomment` was generated by the Peer Workstation Description panel and overwritten by our statement (`ITSCLAB1 Peer Server`).

`Computername` and `Domain` were respectively generated by the Peer Workstation Name and Domain Name panels.

`ReplaceNetAcc=NO` was generated by answering `NO` to the NET.ACC panel.

Answering `NO` to the graphical user interface panel has generated `InstallAdminGUI = REMOVE`. A positive answer would have generated `INSTALL` instead of `REMOVE`.

`ConfigSourceDrive` is set to the chosen drive as an update copy was chosen in the Source Drive panel.

`ConfigTargetDrive` is set to a specific drive if indicated in the Hard Disk panel. These two parameters would then look like (installation on drive C:):

```
ConfigSourceDrive = C
ConfigTargetDrive = C
```

INSTALLIFREQUIRED will install the feature if it is required because some other chosen feature requires it and is not installed already or is an older version.

For features that might be installed already, maybe by another product, it is wise to use INSTALLIFREQUIRED instead of INSTALL, because if the installation program an INSTALL keyword in your response file and that feature/function already is installed with a similar or newer version, the program might end the installation with a bad return code. This is nicely recorded in the log file, but it may take some time to figure out anyway.

The choice of a specific adapter would generate other keywords in the response file. The following figure gives an example of the generated keywords if Adapters 0 and 1 were selected in the LAN Adapter panel (see "Example of an MPTS Response File" on page 226, for more details).

InstallPeerService should be set to REMOVE if only the requester is to be installed.

```
DELETEIBMLAN = Networks<

    net1b =

    net3 =
    net4 =
>

UPDATEIBMLAN = Networks<

    net1 = NETBEUI$,0,LM10,34,100,14
    net2 = tcpbeui$,1,LM10,34,100,14

>

UPDATEIBMLAN = Peer<

    srvcomment = ITSCLAB1 Peer Server

>
```

Figure 61. Example of an OS/2 Peer Services Response File (Part 1 of 2)


```

DELETEIBMLAN = Requester<

    wrknets = net3, net4
    wrkservices = MESSENGER

>

ADDIBMLAN = Requester<

    wrknets = net1, net2
    wrkservices = MESSENGER

>

UPDATEIBMLAN = Requester<

    Computername = ITSCLAB1
    Domain = ITSCAUS

>

    ConfigTargetDrive = C
    ConfigAutoStartLS = Yes
    ConfigSourceDrive = C
    ReplaceNetAcc = NO
    InstallFFST = NO
    InstallAPI = INSTALLIFREQUIRED
    InstallDosLanApi = INSTALLIFREQUIRED
    InstallInstallProgram = INSTALLIFREQUIRED
    InstallPeerService = INSTALL
    InstallRequester = INSTALLIFREQUIRED
    InstallUPM = INSTALLIFREQUIRED
    InstallMSGPopup = INSTALLIFREQUIRED
    InstallGUI = INSTALL
    InstallClipboard = INSTALLIFREQUIRED
    InstallBooks = INSTALLIFREQUIRED
    InstallAdminGUI = INSTALL

```

Figure 62. Example of an OS/2 Peer Services Response File (Part 2 of 2)

7.5 OS/2 LAN Requester Response File

OS/2 Warp Server offers a nice graphical user interface to create response files for OS/2 LAN Requester. Following is a step-by-step approach on how to create a response file.

1. Using the OS/2 Warp Server CD-ROM, change to the \CID\SERVER\IBMLS directory and issue the following command from an OS/2 command line:

```
LANINSTR /SRV
```

2. Click **OK** at the welcome window to get the Easy or Tailored Installation/Configuration window.
3. Click on the **Tailored** button. The Installation Tasks window will appear in which you select the radio button for **Create a requester response file for remote installation** and select **OK**.
4. At the Response File Name window, type a path and response file name and click **OK**. At the File Does Not Exist window, select **Yes** to confirm the specified response file to be created.
5. Answer the next panels the way you would actually install the requester code. When you select **Apply the changes** at the Installation and Configuration window, the response file will be created at the path specified in the previous step.

7.5.1 Example of an OS/2 LAN Requester Response File

The following figure displays a response file created by the OS/2 Warp Server File and Print Services GUI, as illustrated before. Modifications were done in the `UPDATEIBMLAN` section in which NetBEUI is reflected on LAN0 and TCPBEUI on LAN1, since MPTS is configured the way shown in "Example of an MPTS Response File" on page 226.

```

DELETEIBMLAN = Networks<

    net3 =
    net4 =
    netlb =
>
UPDATEIBMLAN = Networks<

    net1 = NETBEUI$,0,LM10,34,100,14
    net2 = tcpbeui$,1,LM10,34,100,14
>
DELETEIBMLAN = Peer<

    srvnets = NETLB,NET3,NET4
>
UPDATEIBMLAN = Peer<

    security = Share
>
ADDIBMLAN = Peer<

    srvnets = NET1,NET2
>
DELETEIBMLAN = Requester<

    wrkservices = PEER
    wrknets = NETLB,NET3,NET4
>
UPDATEIBMLAN = Requester<

    Computername = ITSCWK01
    Domain = ITSCAUS
    useallmem = No
>
ADDIBMLAN = Requester<

    wrkservices = MESSENGER
    wrknets = NET1,NET2
>
    ConfigApplDumpPath = C:\OS2\SYSTEM
    ConfigApplMaxDumps = 32
    ConfigAutoStartFFST = No
    ConfigAutoStartLS = Yes

```

Figure 63. Example of an OS/2 LAN Requester Response File (Part 1 of 2)

```
ConfigDisplayMSG = ON
ConfigDosNumber = 0
ConfigMsgLogName = C:\OS2\SYSTEM\OS2MLOG.DAT
ConfigSourceDrive = None
ConfigSystemDumpPath = C:\OS2\SYSTEM
ConfigSystemMaxDumps = 32
ConfigTargetDrive = C
ConfigWsId = ITSCWK01
ConfigWsSerial1 = 23
ConfigWsSerial2 = 0056MX6
ConfigWsType1 = 6887
ConfigWsType2 = 86B
InstallAPI = INSTALLIFREQUIRED
InstallDosLanApi = INSTALLIFREQUIRED
InstallFFST = INSTALL
InstallInstallProgram = INSTALL
InstallPeerService = INSTALL
InstallRemoteFaultTolerance = INSTALLIFREQUIRED
InstallRequester = INSTALL
InstallUPM = INSTALL
InstallMSGPopup = INSTALL
InstallGUI = INSTALL
InstallClipboard = INSTALL
InstallDesktopIcons = YES
```

Figure 64. Example of an OS/2 LAN Requester Response File (Part 2 of 2)

7.6 TCP/IP Response File

There is no utility program to create a TCP/IP response file. Two sample .RSP files can be found on the OS/2 Warp 4 CD-ROM. The DEFAULT.RSP file is in \CID\IMG\TCPIP directory. A more complete sample file named TCPAPPS.RSP is found when unzipping the \CID\IMG\MPTS\UTILITY\LCU\WARP40.ZIP file.

Description of the TCP/IP keywords is given in the online TCP/IP guide available on any installed OS/2 Warp 4 workstation.

The first part of the response file gives information on the characteristics of the installation and customized values.

In the second part:

- The keyword `INSTALL_NAME` followed by a kit or component name defines which components are to be installed. It will occur more than once according to the number of components you will install.
- The keyword `EXEC` followed by a feature name, a procedure and parameters actually executes the install.

To install TCP/IP, you need to give information to the installation program. For most of the information, it can be supplied in:

- Using environment variables set up in the `CONFIG.SYS` file
- Using keywords in a response file
- Using parameters of the install program

Information specified as parameter has precedence over information in a response file, and this information has precedence over information supplied in environment variable.

Keywords are given in Table 16.

Table 16. TCP/IP Response File Keywords (Part 1 of 2)

Keyword	Description
ATTENDED	NO: Specifies that the installation is to be performed on an unattended basis. The installation window will be displayed at the target workstation, but no action is required of the user.
BOOT_DRIVE	Specifies the drive from which the workstation starts.
CONFIGSYS	NO: Specifies that CONFIG.SYS will not be modified. Modified CONFIG.SYS will be stored as CONFIG.TCP.
CONFIGURE	YES: Specifies that TCP/IP basic configuration is set up. NO: If a previous configuration exists, it will be kept.
DOMAIN	Specifies the domain name of the client workstation.
HOSTNAME	Specifies the host name of the workstation.
HPFS_NEEDED	Specifies the name of components which have to be installed on a HPFS drive.
INETD_SERVICES	Specifies the list of TCP/IP services to be added to the INETD.LST file.
INSTALL_MPTS	Specifies either to install MPTS or not.
LANn={ IPADDR= }	Specifies the IP address of the client workstation for the adapter number n.
LANn={ NETMASK= }	Specifies the subnet mask for the adapter number n.
MPTS_EXE_PATH	Specifies the complete drive and path of the code server that contains the MPTS.EXE file.
MPTS_RSP_FILE	Specifies the fully qualified file name of the MPTS response file.
NAMESERVERn	Specifies the name of a domain name server. Three name servers can be defined for n=1 to 3.
ROUTEn={ ROUTER= }	Specifies the IP address of the router number n.
ROUTEn={ TYPE= }	Specifies the type of router for router number n.

Table 17. TCP/IP Response File Keywords (Part 2 of 2)

Keyword	Description
ROUTEn={ DESTINATION= }	Specifies the router destination address for router n if type is not DEFAULT.
ROUTEn={ METRIC= }	Specifies the hop count for router n.
ROUTEn={ NETMASK= }	Specifies the subnet routing mask for router n.
RSP_PATH	Specifies the path of response file.
TARGET_PATH	Specifies the complete path to which TCP/IP is to be installed.
TCP_SERVICES	Specifies the services to be included in the TCPSTART.COMD file for automatic start-up when TCP/IP initializes. The services are comma separated.
TZ	Specifies the time zone used by workstation.
USE_DDNS	Specifies that DDNS has been installed and will be used to automatically configure the machine's domain name server information during start-up.

7.6.1 Example of a TCP/IP Response File

The following figure shows an example of a TCP/IP response file that was used in our scenarios.

```

INSTALL_WPS = Y

CONFIGURE = Y
ATTENDED = N
LANO = {
    IPADDR = 9.3.1.242
    NETMASK = 255.255.255.0
}
ROUTE0 = {
    ROUTER = 9.3.1.74
    TYPE = DEFAULT
}
HOSTNAME = os2warp4
DOMAIN = itsc.austin.ibm.com
NAMESERVER0 = 9.3.1.74
TARGET_PATH = C:\TCPIP

INSTALL_NAME = BASE 8.65 1 5 "TCP/IP" Base TCP/IP Applications
INSTALL_NAME = INET 2.87 6 7 "TCP/IP" Feature TCP/IP Applications
INSTALL_NAME = DBOX 1.69 7 7 "TCP/IP" DOS\Windows Access
INSTALL_NAME = UMAIL 4.41 8 9 "TCP/IP" UltiMail Lite
INSTALL_NAME = PCOM 9.88 10 13 "TCP/IP" PComm Entry Level for TCP/IP
LANG = EN_US
EXEC = BASE call tcpcoex
EXEC = INET call inetxt
EXEC = DBOX call dboxxt
EXEC = UMAIL call umlitext
EXEC = PCOM call pcomxt
EXEC = PCOM call pcomxt

```

Figure 65. Example of a TCP/IP Response File

7.7 LAN Distance Response File

There is no utility program to set up the LAN Distance response file. An example of this response file can be found when unzipping the \CID\IMG\MPTS\UTILITY\LCU\WARP40.ZIP file.

This file should be edited and updated according to the needs.

7.7.1 LAN Distance Response File Keywords

The following keywords apply to a LAN Distance response file:

Target The drive on which install LAN Distance on the client workstation.

Phone	The phone number of the most commonly used phone number.
LANType	The type of the linked LAN.
	Values can be <code>TokenRing</code> or <code>Ethernet</code> .
Port	The serial port used for connection.
Modem	The name of the <code>.PIF</code> file defining the modem used.
Address	The MAC address to use for the client workstation.

7.7.2 Example of a LAN Distance Response File

The following figure shows a response file for LAN Distance used in our scenarios. LAN Distance response files are stored in the `\CID\RSP\LDREM` directory.

```

Target = C:\
Phone = 555-0127
LANType = TokenRing
Port = COM1
Modem = IBM7855.PIF
Address = 4000E0600213

```

Figure 66. Example of a LAN Distance Response File

7.8 NetFinity Response File

A sample response file called `NETBASE.RSP` can be found in the `\CID\IMG\NETFIN` directory. In this sample file, all the keywords are well documented.

Selection Indicates which object will be selected from the main menu. Values may be `1` for a Passive Client Operation or `2` for an Active Client Operation.

Both selections will install:

- NetFinity Service Manager
- Network communication drivers and configuration program
- Alert manager base program and user interface
- Security manager interface
- Base programs for NetFinity services

In addition, an active will have the following tools to support a

local system management:

System information tool
System profile
Alert manager
System monitor

`InstallTo` This is the directory on a client's target drive into which TME 10 NetFinity will be installed. It will be created if it does not exist. DO NOT include the drive letter in this — that will be assigned at install time.

Example:

```
InstallTo = \NETFIN
```

`ChangeConfig` Installation will update the client CONFIG.SYS file if value is TRUE or 1. FALSE or 0 will cause the necessary changes to be stored in the file CONFIG.NEW in the InstallTo directory.

Example:

```
ChangeConfig = TRUE
```

`DRIVER.X` This will set up the network drivers. The Driver.X fields need to be set to either 1 (driver X is enabled) or 0 (driver X is disabled).

Valid network values are TCPIP, NETBIOS, NETBIOS2, IPX, and SERIPC.

Example:

```
Driver.TCPIP = 1  
Driver.NETBIOS = 1  
Driver.NETBIOS2 = 0  
Driver.IPX = 0  
Driver.SERIPC = 1
```

`ParmN.X` These fields are set according to the various drivers (X). IPX and TCPIP need no parameters.

NETBIOS may need Parm1 — the name of this station, which must be a 1-8 character alphanumeric string.

N must be a number from 1 to 3, and the parms must be in order (meaning if Parm3.NETBIOS exists, it must have both Parm2.NETBIOS and Parm1.NETBIOS before it).

Example:

```
Parm1.NETBIOS = MACHINE1  
Parm1.NETBIOS2 = MACHINE2
```

SERIPC is allowed Parm1 — the name for this station. If not here, a default will be used.

Keyword.N

These fields are the default keywords (for discovery) that will be assigned to this station. N must be a number from 1 to 8.

Example:

```
Keyword.1 = NetFinity
```

NetTimeout

This parameter can be used to set the default

Network Time-Out value for all TME 10 NetFinity applications to a value other than the default of 15 seconds.

This value should only be set if the system is experiencing frequent time-outs that can be attributed to heavy system load and/or heavy network load.

SystemName

The name NetFinity will be recognized as when discovered. The default value is the network address.

Example:

```
SystemName = NetFinity System  
ForceRemoteLogons
```

If this line is uncommented, then all logons from this machine to remote systems will require using a user ID/password logon. (meaning no "default" logons will be used). Only uncomment this line if you wish this to occur.

Example:

```
; ForceRemoteLogons = 1
```

`ServiceAlerts` If this line is uncommented, then any service starting on this machine will send a severity 7 alert to the alertmanager on this machine.

Enabling this will also cause a severity 5 alert to be generated if an attempt is made to access a service which is prevented by security. Only uncomment this line if you want these effects.

Example:

```
; ServiceAlerts = 1
```

7.9 Software Installer Response File

Numerous CID installable programs use Software Installer to install. Their response files comply with the same rules.

Software Installer has a few default values for the response file keywords that it always supports. Additionally, there will be product specific keywords added by the developer. The default keywords and their default usage for the response file are:

<code>OVERWRITE</code>	This keyword specifies if existing files may be overwritten or not. The possible values are <code>YES</code> or <code>NO</code> .
<code>FILE</code>	This keyword points to the target install directory. If you do not provide a value, the default target directory is used — that is the directory the developer decided to be the default installation directory.
<code>AUX1</code>	This keyword specifies a complementary directory which will be used for installation. Usually this directory is a DLL directory.
<code>COMP</code>	This keyword indicates the component to install. There may be multiple occurrences of this keyword if the product has several optional components. The string that is the component's name has to be strictly reported and is case sensitive.
<code>CFGUPDATE</code>	This keyword specifies if <code>CONFIG.SYS</code> and <code>AUTOEXEC.BAT</code> are to be updated automatically during the install process. The possible values are <code>YES</code> , <code>NO</code> and <code>AUTO</code> .

DELETEDBACKUP	This keyword specifies if the product is deleted automatically when a deinstallation takes place. The possible values are YES or NO.
SAVEBACKUP	This keyword specifies if a backup version of the product is automatically created when the product is updated. The possible values are YES or NO.

Attention

These are the default keywords proposed by the Software Installer product. The developer of the product might have changed these defaults. Please check the product information. If you do not find any hint to these keywords, you can assume that they are used the way described here.

7.10 Network Signon Coordinator/2 Response File

This product has only one component. A typical response file is shown below:

```
FILE = C:\NSC
AUX1 = C:\NSC\DLL
comp=Network SignON Coordinator/2
cfgupdate=auto
overwrite=no
savebackup=no
deletebackup=no
```

7.11 SystemView Agent Response File

This product has only one component. In addition to the normal Software Installer keywords, it has specific keywords. A minimum response file is shown below:

```
FILE = C:\SVCA
COMP = DMI and SNMP
CFGUPDATE = AUTO
DELETEDBACKUP = NO
OVERWRITE = YES
SAVEBACKUP = NO
NETVIEW_AGENT_IP_TRANSPORT = NO
NETVIEW_AGENT_NETBIOS_TRANSPORT = YES
NETVIEW_AGENT_IPX_TRANSPORT = NO
```

7.11.1 Specific Keywords

All these keywords have default values; so most of them have do not have to be defined.

NETVIEW_AGENT_IP_TRANSPORT	(default YES)
NETVIEW_AGENT_NETBIOS_TRANSPORT	(default YES)
NETVIEW_AGENT_IPX_TRANSPORT	(default NO)
	Enable or not the transport layer protocol
NETWORK_SYSTEM_DESCRIPTION	(default: SystemView Agent for OS/2)
	Information about the OS/2 agent
NETVIEW_SYSTEM_CONTACT	(default: Jane Doe)
	The name of the contact person for the OS/2 agent
NETVIEW_SYSTEM_NAME	(default: PWS206)
	The host name of the OS/2 agent
NETVIEW_SYSTEM_LOCATION	(default: Bldg 001)
	Information on the physical location if the OS/2 agent
NETVIEW_AUTHENTICATION_TRAPS	(default: ENABLE)
	Enables the OS/2 agent for sending authentication failure traps
NETVIEW_COMMUNITY_NAME_n	(default: publicnbi for n=1 and public for n=4)
	Community names used to verify SNMP requests from remote SNMP agents. n can be 1 to 20.
NETVIEW_COMMUNITY_PROTOCOL_n	(default: netbios for n=1 and udp for n=4)
	Transport protocol to use with community name definition n. Valid values are: udp, netbios and ipx.

NETVIEW_COMMUNITY_ACCESS_TYPE	(default: write)
	Access type used. Valid values are: read, write and trapdest.
NETVIEW_COMMUNITY_ADDRESS_n	Address from which to process requests or to which send traps. Address depends on the transport type.
NETVIEW_COMMUNITY_NETWORK_MASK	Only valid if protocol is UDP.
	Network mask to apply to the originating IP address of an incoming SNMP request.

7.12 Mobile File Sync (MFS) Response File

This product has only one component. A typical response file is shown below:

```
file=c:\mfs
comp=Mobile File Sync program
cfgupdate=auto
overwrite=no
savebackup=no
deletebackup=no
```

7.13 Database 2 for OS/2 Response File

Database 2 for OS/2 includes a program to generate response files. This program, DB2RESP.EXE, is a PM program which will allow you to generate a basic response file. This response file may need to be customized afterwards. DB2RESP.EXE can be found in the image directories of DB2 server, DB2 single-user and DB2 Client Application Enabler.

To generate a basic response file for the Server version, perform the following steps:

1. Change to the directory CID\IMG\DB2V2 (assuming this is the DB2 single-user image directory)
2. Invoke DB2RESP.
3. The first panel is called IBM DB2 Create Installation Response File. Open the Product drop-down list and select **DB2ServerProd**. This will update the Optional Components list below.
4. Select all the components you want to install from this list.
5. Type the name of the file directory or accept the default C:\SQLLIB

6. Specify CONFIG.SYS and BACKUP options or accept the preset defaults.
7. Select **Save**.
8. Specify the name of the response file and a directory to place it in. When finished click on **OK**.
9. Click on **Exit**.
10. You have now a basic response file. A DB2 CAE (Client Application Enabler) response file is displayed in Figure 67 on page 250.

The generation of another response file for a different product set is similar. The only difference is the selection of a different product in step 3.

The generated response file includes the usual Software Installer keywords, including the `COMP` keyword of each of the selected components. In addition, there is a lot of specific keywords for DB/2. There are all generated as comments. Some of them have to be activated. For example, for DB2 server, `DB2COMM` to choose the type of communication or `LICENSEKEY` to give the license number.

For detailed information on DB/2 response files, please refer to the documentation that comes with the product. Also make sure to check the README file for the latest updates.

7.13.1 Use of Client-Specific Response File for Database 2 Installation

Except when installing Database 2 for OS/2 stand-alone workstations, client-specific response files are needed.

Consider how you can group users with similar requirements and provide one default model for each group. There are a couple of options for Database 2 for OS/2 on how to provide client-specific response files:

1. You can make one complete response file for each client.

It is rare that the Database 2 for OS/2 installation is changed, but it is not unusual that the configuration of the clients are changed after installation. Configuration changes usually are done to enhance performance for the database applications run in a specific business environment.
2. You can use the `INCLUDE` keyword to include a default response file and then further down in the client's response file, set the keywords you want to be specific.
3. You can use the `DBModelCFG` keyword and use an installed configuration as model. On an installed system the database manager configuration information is kept in the `\SQLLIB\SQLSYSTEM` file.

INCLUDE Keyword in Database 2 for OS/2 Response File

When using an `INCLUDE` in a Database 2 for OS/2 response file, the processing of the keywords in the included file is done immediately. This matches the behavior of the `IncludeInLine` keyword for OS/2.

A fully qualified path is needed to the include file.

Assuming that you have a common response file, `DB2SU.RSP`, and you only want to change the database workstation name, a client-specific response file could look like:

```
; DB2/2 V2.11 Single-User Version response file
Include=X:\RSP\DB2SU\DB2SU.RSP
DBWorkstationName=DB2WS
```

The following figure displays a DB2 CAE (Client Application Enabler) response file created by the `DB2RESP.EXE` utility (provided with the DB2 package).

```

FILE                = C:\SQLLIB
COMP                = Client Application Enabler
OVERWRITE           = YES
CFGUPDATE           = MANUAL
SAVEBACKUP          = NO

*DIAGLEVEL          = 0 - 4
*DIAGPATH           = char(215)
*DIR_CACHE          = YES or NO
*DRDA_HEAP_SZ       = 16 - 60000
*NNAME              = char(8)
*RQRIOBLK           = 4096 - 65535
*SYSADM_GROUP       = char(8)
*SYSCTRL_GROUP      = char(8)
*SYSMAINT_GROUP     = char(8)
*TM_DATABASE        = char(8)
*TP_MON_NAME        = char(19)
*DB2BQTIME          = >= 1
*DB2BQTRY           = >= 0
*DB2CHKPTR          = ON or OFF
*DB2DBDFT           = char(8)
*DB2IQTTIME         = >= 1
*DB2OPTIONS         = char(1024)
*DB2RQTIME          = >= 1
*DB2ACCOUNT         = char(25)
*IVLSQLLIMIT        = >= 0
*IVLSQLRTYPE        = CS, RR, or UR
*IVLSQLUTYPE        = EXCLUSIVE or SHARE

```

Figure 67. Database 2 Client Application Enabler Response File

7.14 MQ Series Response File

MQ series response file is a standard Software Installer response file. A sample response file called AMQISAM2.RSP can be found in the image directory of MQ Series.

This file has to be customized to choose the components to install.

Figure 68 is an example of such a response file.

```

*****
FILE=c:\mqm
*****
WORK=c:\mqm
*****
COMP=OS/2 Client
COMP=Base product and Server
COMP=On-line Information
COMP=Toolkit
COMP=DOS Client
COMP=Windows Client
*COMP=AIX Client
*****
CFGUPDATE=AUTO
*****
OVERWRITE=NO
*****
SAVEBACKUP=NO
*****
DELETEBACKUP=NO
*****

```

Figure 68. Example of a MQ Series Response File

7.15 Transaction Server (CICS) Response File

There are two response files to be considered: one response file for the server and one for the client.

7.15.1 Server Response File

A sample response file called CICSOS2.RSP can be found in the image directory of CICS. Remember that the transaction server CD-ROM has a different directory for each supported language.

The response file is a Software Installer response file. In addition to the standard keywords, the following keywords have to be coded:

APPLID	Specifies the application identification that will be used by the installed CICS system.
	This value is used to identify the system for NetBIOS communications.
SYSID	Specifies the system identification that will be used to identify this CICS system.

LICENSES	Specifies the number of concurrent users for which licenses have been bought.
LICENSEFILE	Specifies the fully qualified path and name of the license file on the client machine.

Refer to the product's documentation to get information about licenses.

Note: CICS response file does not allow the customization of CICS tables. This must be done after installation using the utilities in the CICS runtime subdirectory (CICSREAD.COMD and CICSLOAD.COMD).

7.15.2 CICS Client Response File

A CICS client response file model can be found in the image directory of CICS client. Its name is CICSCLI.RSP.

The response file is a Software Installer response file. In addition to the standard keywords, the following keywords have to be coded:

CICSCLINI	Specifies the name of a customized initialization file.
-----------	---

If there is no such file, the value must be set to an asterisk (*).

See next page for a sample CICSCLI.INI file.

CICSCLI.INI File

This file defines the communication protocol to communicate with the transaction server. The default file is customized for a NetBIOS communication. If another protocol is to be used, it is necessary to change the default as well as change the net name of the client. A CICSCLI.INI file for TCP/IP would look like :

```
Client = *                ; Auto-install client on the server
  MaxServers = 1          ; Only allow one server connection
  MaxRequests = 20        ; Limit the maximum server interaction
  MaxBufferSize = 32     ; Allow for a 32K maximum COMMAREA
  LogFile = CICSCLI.LOG   ; Set the error log file name
  TraceFile = CICSCLI.TRC ; Set the trace log file name
  DosMemory = 48          ; The DOS client's memory pool size

Server = CICSTCP          ; Arbitrary name for the server
  Description = TCP/IP Server ; Arbitrary description for the server
  Protocol = TCPIP        ; Matches with a Driver section below
  NetName = 10.45.99.99   ; The server's TCP/IP address
  Port = 0                ; Use the default TCP/IP CICS port

Driver = TCPIP            ; Matches the Server's Protocol value
  DriverName = CCLIBMIP   ; Use the IBM TCP/IP communications DLL
```

CICSCOLINI Specifies the name of an initialization file to set the colors of the CICS terminal emulator.

If it does not exist, the value must be set to an asterisk (*).

CICSKEYINI Specifies the name of an initialization file to set the keys of the CICS terminal emulator.

If it does not exist, the value must be set to an asterisk (*).

MESSAGELANG Specifies the language in which messages are sent to the user.

Figure 69 shows an example of CICS client response file. In this example, the CICSCLI.INI file used to define the communication protocol is located in the same directory as the response file.

```
CFGUPDATE = AUTO
COMP = CICS Client runtime system
*COMP = CICS Client programming samples
*COMP = CICS Gateway for Lotus Notes
*COMP = CICS Internet Gateway
COMP = CICS Client documentation
FILE = C:\CICSCLI
OVERWRITE = YES
DELETEBACKUP = NO
CICSCLIINI = \CID\RSP\CICS\CICSCLI.INI
CICSCLINI = *
CICSKEYINI = *
MESSAGELANG = FR
```

Figure 69. Example of a CICS Client Response File

7.16 Feature Install (CLIFI) Response File

The following section introduces Feature Installer techniques with the focus on installing objects through response files. As mentioned in “New Keywords in OS/2 Warp 4” on page 204, Feature Installer is used to install additional OS/2 Warp 4 components, such as Java, Developer Extensions and so forth, and BonusPak features, such as IBM Works, VoiceType and so forth. Find a more common approach of how Feature Installer uses a particular response file which might help you to get your software installed through Feature Installer rather than a product software installer.

7.16.1 Creating and Reading a Response File

The install object response file is an ASCII file that contains an entire install object hierarchy. This file includes all of the instance data from each install object in the hierarchy as well as information that defines the hierarchy structure.

A response file is created by selecting **Response file** and then **Save** from the install object's pop-up menu. A Save Response File dialog is displayed where you can specify the drive, path and file name of the response file. The response file will contain the install object hierarchy beginning from the install object where the Response File Save option was selected.

For example, if **Response File Save** is selected from the top install object, then the entire object hierarchy will be saved to the response file. If

Response File Save is selected from one of the subfeature install objects, then only that subfeature's hierarchy will be saved.

A response file can be used to create an entire install object hierarchy. The `CLIFI` utility provided with the Feature Install Developer's Toolkit will create an install object hierarchy from a response file. Also, an existing install object's settings and underlying hierarchy can be loaded by selecting **Response File** and then **Read** from the object's pop-up menu. The Read Response File dialog is displayed where you can specify the drive, path and file name of the response file.

7.16.2 Response File Example

Following is an example of a response file. Comments can be placed anywhere in the response file. If # is the first character in the line, the rest of the line is treated as a comment. If you edit the response file, be sure to use an editor that preserves tabs.

```
# Sample response file
TopObjectID=MYFID
MYFID=(
  SubfeatureID=MYFID Child
  ObjectTitle[0]=FI Object
  Mode=Development
  PackageTitle=MPT
  CompanyName=IBM
  VersionNumber=1.45
  VersionDate=03-19-97

  Variable=(
    Name=variable1
    Description=Dsc. of var1
    Value=Default_value
    ValidationExit=(
      CallTime=19
      ExitType=2
      FilePath=DLLNAME.DLL
      ExitData=UserValidate
      SessionVisible=1
    )

    ResolutionExit=(
      CallTime=18
      ExitType=2
      FilePath=CMDRES.CMD
      ExitData=-pcmd1 -pcmd2
      SessionVisible=1
    )
  )
)
```

```

    )
)

Dependency=(
    FeatureID=ConditionID
    ActionFeatureID=ResultID
    WhenFeatureID=CauseID
    ResolveTime=1
    Condition=1
    Action=1
    ActionText[0]=Text Text Text
    Setting=SettingKey
    Value=SettingValue
    ResolveNow=1
    CaseSensitive=0
)

UserExits=(
    ExitType=1
    FilePath=DLLNAME.DLL
    ExitData=-pcmdl -pcmd
    SessionVisible=1
    CallTime=1
    ObjectCreation=(
        ClassName=WP_CLASS
        ObjectTitle[0]=My Program
        SetupString=EXENAME=exefile.exe^;PARAMS=param
        Location=<WP_FOLDERNAME>
        Flags=0
    )

    ClassRegistration=(
        ClassName=WP_CLASS
        DLLName=regclass.dll
        ClassReplace=WP_OLD
    )

    MediaSet=(
        SetName=MediaSetName
        Media=(
            Type=2
            Capacity=48
            ClusterSize=34
            Description=CD
            MediaPath=D:\
            ReservedSpace=1000
        )
    )
)

```



```

MediaFinishedExit=(
    CallTime=20
    ExitType=1
    FilePath=DLLNAME.DLL
    ExitData=-pcmd1 -pcmd2
    SessionVisible=1
)
)

AsciiFile=(
    CaseSensitive=1
    UseGrep=0
    FileName=config.sys
    EditLine=Line To Edit
    SearchLine=set path
    NewLine=rem set path
    SearchFound=2
    SearchNotFound=2
    AddBeforeLine=beforeline
    AddAfterLine=afterline
    Delimiter=*
    SubstrChk=1
    DelimChk=1
    Action=1
    EditLineOcc=2
    SrcLineOcc=1
    AddBLineOcc=1
    AddALineOcc=1
    MatchPosition=2
    NoMatchPosition=2
)

Os2PrfIni=(
    FileName=d:\path\os2.ini
    Application=ApplicationName
    Key=KeyName
    Value=KeyValue
    IniType=1
)

WinOs2Ini=(
    FileName=d:\winpath\win.ini
    Application=WinAppName
    Key=WinKeyName
    Value=WinKeyValue
    IniType=2
)

```

```

    )

File=(
    EAMediaIndex=0
    Source=S:\srcpath\srcname
    SourcePath=S:\srcpath
    SourceFileName=srcname
    SourceChecksum=0
    SourceEASize=0
    CBFile=0
    CBFileAlloc=0
    AttrFile=0
    TargetPath=T:\targetpath
    TargetFileName=srcname
    MediaPath=mediapath
    MediaFileName=srcname
    MediaNumber=0
    Flags=0
    TargetAttrib=3

    Restriction=(
        MediaSet=MyMediaSet
        MediaNumber=1
    )
)

Prerequisite=(
    FeatureID=PrereqFeature
)
)

```

7.16.3 FI Response File Keywords

The following list describes each of the keywords that can be used in a Feature Installer response file. The keywords listed include sample values as shown in “Response File Example” on page 255.

TopObjectID=MYFID	Feature ID that identifies the top install object. <i>Valid Values:</i> Any unique string. The string can contain spaces but must not contain any periods.
MYFID=(Current feature ID marking the beginning of the install object description. This type of block can appear multiple times.

	<i>Valid Values:</i> Any previously defined feature ID.
SubfeatureID=MYFID Child	Subfeature ID. <i>Valid Values:</i> Any defined feature ID (can appear multiple times if more than subfeature exists, or can be omitted if there are no subfeatures).
ObjectTitle[0]=FI Object	Object title[line# - 1]. <i>Valid Values:</i> Any ordered set of strings.
Mode=Development	State of the object. <i>Valid Values:</i> Development, User, or Inventory.
PackageTitle=MPT	Package title. <i>Valid Values:</i> Any string.
CompanyName=IBM	Company name. <i>Valid Values:</i> Any string.
VersionNumber=1.45	Version number. <i>Valid Values:</i> Any string.
VersionDate=03-19-97	Version date. <i>Valid Values:</i> Any string.
Variable=(Marks the beginning of a variable description. This type of block can appear multiple times. <i>Valid Values:</i> No value associated with this keyword.
Name=var1	Variable name. <i>Valid Values:</i> Any string (the string can contain spaces).
Description=Description of var1	Description of the variable. <i>Valid Values:</i> Any string.
Value=Default_value	Default value for the variable. <i>Valid Values:</i> Any string.

ValidationExit=(Marks the beginning of a variable validation user exit description, <i>Valid Values:</i> No value associated with this keyword.
CallTime=19	Determines when the user exit is called (can be omitted but can not be specified with an invalid value). <i>Valid Values:</i> 19 Variable validation user exit 18 Variable resolution user exit
ExitType=2	Type of user exit. <i>Valid Values:</i> 0 DLL (default; can be omitted) 1 EXE 2 CMD
FilePath=DLLNAME.DLL	User exit location. <i>Valid Values:</i> Fully qualified path name.
ExitData=UserValidate	Details for the user exit. <i>Valid Values:</i> For exit type 0, this is the name of the procedure in the DLL that is being called. For exit type 1, this is the EXE file parameters. For exit type 2, this is the CMD parameters
SessionVisible=1	If the session is visible or minimized (for ExitType=1 or 2 only). <i>Valid Values:</i> 0 minimized 1 visible to user
ResolutionExit=(Marks the beginning of a variable resolution user exit description. <i>Valid Values:</i> No value associated with this keyword.
CallTime=18	Determines when the user exit is called (can be omitted). <i>Valid Values:</i> 19 Variable validation user exit 18 Variable resolution user exit
ExitType=2	Type of user exit. <i>Valid Values:</i>

	<ul style="list-style-type: none"> 0 DLL (default) 1 EXE 2 CMD
FilePath=CMDRES.CMD	<p>User exit location.</p> <p><i>Valid Values:</i> Fully qualified path.</p>
ExitData=-pcmd1 -pcmd2	<p>User exit details.</p> <p><i>Valid Values:</i> For exit type 0, the name of the procedure in DLL that is being called. For exit type 1, the EXE file parameters For exit type 2, the CMD file parameters</p>
SessionVisible=1	<p>If the session is visible or minimized (for ExitType=1 or 2 only).</p> <p><i>Valid Values:</i> 0 minimized 1 visible to user</p>
Dependency=(<p>Marks the beginning of a dependency description. This type of block can appear multiple times.</p> <p><i>Valid Values:</i> No values associated with this keyword.</p>
FeatureID=ConditionID	<p>ID of the feature for which the state is examined to perform a dependency action. If the FeatureID in a dependency is blank, the current object's feature ID will be used.</p> <p><i>Valid Values:</i> Any previously defined feature ID.</p>
ActionFeatureID=ResultID	<p>ID of the feature for which the state is examined to perform a dependency action.</p> <p><i>Valid Values:</i> Any previously defined feature ID.</p>
WhenFeatureID=CauseID	<p>ID of the feature of which the state is examined to evaluate the need to check the existence of the dependency condition.</p> <p><i>Valid Values:</i> Any previously defined feature ID.</p>
ResolveTime=1	<p>Determines when the condition of the feature defined by When FeatureID is evaluated. <i>Valid Values:</i></p>

	<ul style="list-style-type: none"> 0 selected 1 deselected 2 installed 3 uninstalled
Condition=1	<p>Determines the condition of the feature defined by <code>FeatureID</code> that has to be met for the dependency action to be performed.</p> <p><i>Valid Values:</i></p> <ul style="list-style-type: none"> 0 exists (default) 1 does not exist 2 selected 3 is not selected 4 equal 5 not equal 6 less than 7 less or equal 8 greater than 9 greater or equal 10 contains 11 is part of
Action=1	<p>Determines what dependency action is to be performed if all conditions are met.</p> <p><i>Valid Values:</i></p> <ul style="list-style-type: none"> 0 select feature 1 deselect feature 2 uninstall feature 3 display warning 4 display error 5 set variable
ActionText[0]=Text Text Text	<p>Text to be displayed for the warning (<code>Action=3</code>) or for the error (<code>Action=4</code>) [<code>line#-1</code>] (for <code>Action=3</code> or <code>Action=4</code> only).</p> <p><i>Valid Values:</i></p> <p>Any ordered set of strings.</p>
Setting=SettingKey	<p>Key to be compared to the value defined by <code>Value</code> key in the feature defined by <code>ActionFeatureID</code> (the true comparison is defined by <code>Condition</code> key). This is valid for <code>Condition=4</code>, <code>Condition=5</code>, <code>Condition=6</code>, <code>Condition=7</code>, <code>Condition=8</code>, and <code>Condition=9</code> only.</p>

Value=SettingValue	<p><i>Valid Values:</i> Any key valid for this feature.</p> <p>Value to be compared to the value of the key. The key is defined by the <code>Setting</code> key in the feature, which is defined by <code>ActionFeatureID</code> (the true comparison is defined by <code>Condition</code> key). This is valid for <code>Condition=4, 5, 6, 7, 8, and 9</code> only.</p> <p><i>Valid Values:</i> Any value compatible to the <code>Setting</code> key type.</p>
ResolveNow=1	<p>If a <code>Value</code> key setting contains variables, this key determines if they are resolved before the <code>Value</code> is assigned to the <code>Setting</code>. Otherwise, the <code>Setting</code> is assigned the value with variable references in it (for <code>Action=5</code> only).</p> <p><i>Valid Values:</i> 0 variables resolved at some later point (default) 1 variables resolved before the value is assigned</p>
CaseSensitive=0	<p>Determines if the comparison defined by <code>Condition</code> is case sensitive (for <code>Condition=4, 5, 6, 7, 8, and 9</code> only).</p> <p><i>Valid Values:</i> 0 case insensitive comparison 1 case-sensitive comparison (default)</p>
UserExits=(<p>Marks the beginning of a user exit description. This type of block can appear multiple times.</p> <p><i>Valid Values:</i> No value associated with this keyword.</p>
ExitType=1	<p>User exit type.</p> <p><i>Valid Values:</i> 0 DLL (default) 1 EXE 2 CMD</p>
FilePath=DLLNAME.DLL	<p>User exit location.</p> <p><i>Valid Values:</i> Fully qualified path name.</p>

ExitData=-pcmd1 -pcmd2	<p>User exit details.</p> <p><i>Valid Values:</i></p> <p>For exit type 0, name of the procedure in DLL that is being called.</p> <p>For exit type 1, EXE file parameters.</p> <p>For exit type 2, CMD file parameters.</p>
SessionVisible=1	<p>Specifies if the session is visible or minimized (for ExitType=1 or 2 only).</p> <p><i>Valid Values:</i></p> <p>0 minimized (default)</p> <p>1 visible to user</p>
CallTime=1	<p>Determines when the user exit is called.</p> <p><i>Valid Values:</i></p> <p>0 after awakening (default)</p> <p>1 before sleeping</p> <p>2 before install</p> <p>3 before file transfer</p> <p>4 after file transfer</p> <p>5 before configuration updates</p> <p>6 after configuration updates</p> <p>7 before object creation</p> <p>8 after object creation</p> <p>9 after install</p> <p>10 before uninstall</p> <p>11 before file deletion</p> <p>12 after file deletion</p> <p>13 before configuration entry removal</p> <p>14 after configuration entry removal</p> <p>15 before object deletion</p> <p>16 after object deletion</p> <p>17 after uninstall</p>
ObjectCreation=(<p>Marks the beginning of the description of an object to be created. This type of block can appear multiple times.</p> <p><i>Valid Values:</i></p> <p>No values associated with this keyword.</p>
ClassName=WP_CLASS	<p>Class name for the object.</p> <p><i>Valid Values:</i></p> <p>Any registered class name.</p>

ObjectTitle[0]=My Program	Object title[line#-1]. <i>Valid Values:</i> Any ordered set of strings.
SetupString=EXENAME=exefile.exe^;PARAMS=param	Setup string for the object. <i>Valid Values:</i> Setup string without any spaces consisting of pairs Key=Value, delimited by ^; (caret-semicolon combination). For a list of valid keys and appropriate values, refer to the <i>Workplace Shell Programming Guide</i> .
Location=<WP_FOLDERNAME>	Location of the object on the Desktop. <i>Valid Values:</i> Object ID for a folder. If folder does not exist, the object is placed on the Desktop (<WP_DESKTOP>).
Flags=0	Object creation flags that determine action performed if the object already exists. <i>Valid Values:</i> 0 CO_FAILIFEXISTS 1 CO_REPLACEIFEXISTS (default) 2 CO_UPDATEIFEXISTS 3 CO_PRESERVEOLD (This flag can be used to create only Workplace Shell objects on OS/2 Warp 4 systems.)
ClassRegistration=(Marks the beginning of the description of a class to be registered. This type of block can appear multiple times. <i>Valid Values:</i> No values associated with this keyword.
ClassName=WP_CLASS	Class name <i>Valid Values:</i> Class name defined in DLL determined by DLLName key. For a full description of this key, refer to the description of <i>WinRegisterObjectClass</i> in the <i>Workplace Shell Programming Guide</i> .
DLLName=regclass.dll	Name of the DLL containing the definition of the class. <i>Valid Values:</i> Fully qualified path name. For a description of

ClassReplace=WP_OLD	valid keys and their appropriate values, refer to the description of <i>WinRegisterObjectClass</i> in the <i>Workplace Shell Programming Guide</i> . Class to replace with the class determined by <i>ClassName</i> key (can be omitted if no class has to be replaced). <i>Valid Values:</i> Registered class name. For a description of valid keys and their appropriate values, refer to the description of <i>WinRegisterObjectClass</i> in the <i>Workplace Shell Programming Guide</i> .
MediaSet=(Marks the beginning of a media set description. This type of block can appear multiple times. <i>Valid Values:</i> No values associated with this keyword.
SetName=MediaSetName	Media set name. <i>Valid Values:</i> A string without any spaces.
Media=(Marks the beginning of a media description. <i>Valid Values:</i> No value associated with this keyword.
Type=2	Media type. <i>Valid Values:</i> 1 diskette (default) 2 CD 4 hard drive
Capacity=48	Media capacity (for <i>Type=2</i> only). <i>Valid Values:</i> Any number.
ClusterSize=34	Cluster size (for <i>Type=2</i> only). <i>Valid Values:</i> Any number.
Description=CD	Media description. <i>Valid Values:</i> Any string (can be omitted for <i>Type=1</i>)
MediaPath=D:\	Media path. <i>Valid Values:</i> Qualified path (can be omitted for <i>Type=1</i>).

ReservedSpace=1000	Reserved space on the media in bytes. <i>Valid Values:</i> Any string (can be omitted if 0).
MediaFinishedExit=(Marks the beginning of the description of the user exit to be run if there is no more space on the media. <i>Valid Values:</i> No value associated with this keyword.
CallTime=20	Indicates that the user exit processes the finished media case. <i>Valid Values:</i> 20
ExitType=1	User exit type. <i>Valid Values:</i> 0 DLL (default) 1 EXE 2 CMD
FilePath=DLLNAME.DLL	User exit location. <i>Valid Values:</i> Fully qualified path name.
ExitData=-pcmd1 -pcmd2	User exit details. <i>Valid Values:</i> For exit type 0, name of the procedure in DLL that is being called. For exit type 1, EXE file parameters. For exit type 2, CMD file parameters.
SessionVisible=1	Specifies if the session is visible or minimized (for ExitType=1 or 2 only). <i>Valid Values:</i> 0 minimized (default) 1 visible to user
AsciiFile=(Marks the beginning of an ASCII configuration file change description. This type of block can appear multiple times. <i>Valid Values:</i> No values associated with this keyword.
CaseSensitive=1	Determines if search for the entity to be updated is case sensitive. <i>Valid Values:</i>

	0 case insensitive (default; can be omitted) 1 case-sensitive
UseGrep=0	If <code>grep</code> is to be used to search for the entity to be updated. <i>Valid Values:</i> 0 do not use <code>grep</code> 1 use <code>grep</code> (default; can be omitted)
FileName=config.sys	Configuration file name. <i>Valid Values:</i> Qualified file name.
EditLine=Line To Edit	Line to edit (for <code>Action=1</code> only). <i>Valid Values:</i> Any string.
SearchLine=set path	Line to search for <code>Action=0</code> and a substring in searched lines to scan for <code>Action=1</code> . <i>Valid Values:</i> Any string.
NewLine=rem set path	Line to replace string with. <i>Valid Values:</i> Any string.
SearchFound=2	Determines action to be performed if the string defined by <code>SearchLine</code> is found. <i>Valid Values:</i> 1 replace with the string defined by <code>NewLine</code> (default; can be omitted) 2 take no action
SearchNotFound=2	Determines action to be performed if the string defined by <code>SearchLine</code> is not found. <i>Valid Values:</i> 1 add new line to beginning of file 2 add new line to end of file 4 add before or after other lines 8 take no action
AddBeforeLine=beforeline	Line before which the line defined by <code>SearchLine</code> is to be inserted (for <code>SearchNotFound=4</code> only). <i>Valid Values:</i> Any string.

AddAfterLine=afterline	Line after which the line defined by SearchLine is to be inserted (for SearchNotFound=4 only). <i>Valid Values:</i> Any string.
Delimiter=*	If the value is treated as a group of delimited substrings, the delimiter for these substrings (for Action=1 only). <i>Valid Values:</i> Any string without spaces.
SubstrChk=1	Determines if the value is treated as a group of delimited substrings (for Action=1 only). <i>Valid Values:</i> 0 no 1 yes
DelimChk=1	Determines if there is a delimiter after last entry (for Action=1 only). <i>Valid Values:</i> 0 no 1 yes
Action=1	Configuration file editing action to perform. <i>Valid Values:</i> 0 replace a line with another line (default; can be omitted) 1 edit a selected line
EditLineOcc=2	Determines which occurrence of the string defined by EditLine in the configuration file is looked for (for Action=1 only). <i>Valid Values:</i> 0 first (default; can be omitted) 1 last 2 every
SrcLineOcc=1	Determines which occurrence of the string defined by SearchLine in the configuration file is looked for. <i>Valid Values:</i> 0 first (default; can be omitted) 1 last 2 every

AddBLineOcc=1	Determines which occurrence of the string defined by AddBeforeLine in the configuration file is looked for (for SearchNotFound=4 only). <i>Valid Values:</i> 0 first (default; can be omitted) 1 last 2 every
AddALineOcc=1	Determines which occurrence of the string defined by AddAfterLine in the configuration file is looked for (for SearchNotFound=4 only). <i>Valid Values:</i> 0 first (default; can be omitted) 1 last 2 every
MatchPosition=2	Determines action to be performed if the position to insert the string defined by SearchLine is found (for SearchNotFound=4 only). <i>Valid Values:</i> 1 replace with the string defined by NewLine (default; can be omitted) 2 take no action
NoMatchPosition=2	Determines action to be performed if the position to insert the string defined by SearchLine is not found (for SearchNotFound=4 only). <i>Valid Values:</i> 1 add new line to beginning of file 2 add new line to end of file 8 take no action
Os2PrfIni=(Marks the beginning of an OS/2 .INI configuration file change description. This type of block can appear multiple times. <i>Valid Values:</i> No values associated with this keyword.
FileName=d:\path\os2.ini	Configuration file name. <i>Valid Values:</i> Qualified file name.
Application=ApplicationName	Application whose OS/2 .INI file is to be changed.

	<p><i>Valid Values:</i> Any string.</p>
Key=KeyName	<p>Name of the key of which the value is to be changed for the application determined by the Application keyword.</p> <p><i>Valid Values:</i> Any string without any spaces.</p>
Value=KeyValue	<p>Value to be set for the key determined by the Key keyword.</p> <p><i>Valid Values:</i> Any appropriate value.</p>
IniType=1	<p>Set to 1 for the OS/2 .INI configuration files.</p> <p><i>Valid Values:</i> 1 is the only valid value.</p>
WinOs2Ini=(<p>Marks the beginning of a WIN-OS/2 .INI configuration file change description. This type of block can appear multiple times.</p> <p><i>Valid Values:</i> No values associated with this keyword.</p>
FileName=d:\path\os2.ini	<p>Configuration file name.</p> <p><i>Valid Values:</i> Qualified file name.</p>
Application=WinAppName	<p>Application whose WIN-OS/2 .INI file is to be changed.</p> <p><i>Valid Values:</i> Any string.</p>
Key=WinKeyName	<p>Name of the key of which the value is to be changed for the application determined by the Application keyword.</p> <p><i>Valid Values:</i> Any string without any spaces.</p>
Value=KeyValue	<p>Value to be set for the key determined by the Key keyword.</p> <p><i>Valid Values:</i> Any appropriate value.</p>
IniType=2	<p>Set to 2 for the WIN-OS/2 .INI configuration files.</p> <p><i>Valid Values:</i> 2 is the only valid value.</p>

File=(Marks the beginning of the description of the file to be transferred during installation. This type of block can appear multiple times. <i>Valid Values:</i> No values associated with this keyword.
EAMediaIndex=0	Indicates that the media supports extended attributes. <i>Valid Values:</i> Filled in automatically during packaging.
Source=S:\srcpath\srcname	File path of the source. <i>Valid Values:</i> Fully qualified file name.
SourcePath=S:\srcpath	Location of the source path. <i>Valid Values:</i> Qualified path name.
SourceFileName=srcname	File name of the source. <i>Valid Values:</i> Qualified file name.
SourceChecksum=0	Checksum for the file on the source. <i>Valid Values:</i> Set automatically.
SourceEASize=0	EA (Extended Attributes) size for the file on the source. <i>Valid Values:</i> Set automatically.
CBFile=0	Current number of bytes used for the file on the source. <i>Valid Values:</i> Set automatically.
CBFileAlloc=0	Current number of bytes allocated for the file on the source. <i>Valid Values:</i> Set automatically.
AttrFile=0	Not used.
TargetPath=S:\targetpath	File path of the target. <i>Valid Values:</i> Qualified path name.

TargetFileName=targetname	File name of the target. <i>Valid Values:</i> Qualified file name.
MediaPath=mediapath	File path on the media. <i>Valid Values:</i> Qualified path name.
MediaFileName=srcname	File name on the media. <i>Valid Values:</i> Qualified file name.
MediaNumber=0	Media number in the set. <i>Valid Values:</i> Number between 0 and n-1 where n is the number of media defined in the media set.
Flags=0	Not used.
TargetAttrib=3	Attributes on the file on the target. <i>Valid Values:</i> 0 - 7 in binary representation where the first digit stands for system file (0 - off, 1 - on), second digit stands for hidden file (0 - off, 1 - on), and third digit stands for read-only file (0 - off, 1 - on). The values are: 0 regular file 1 read-only file 2 hidden file 3 hidden read-only file 4 system file 5 read-only system file 6 hidden system file 7 hidden read-only system file
Restriction=(Marks the beginning of a media restriction for the file section. This type of block can appear multiple times. <i>Valid Values:</i> No value associated with this keyword.
MediaSet=MyMediaSet	The name of the media set for which the file is restricted to be placed. <i>Valid Values:</i> Any previously defined media set name.

MediaNumber=1	The ordinal number of the media in the media set determined by the MediaSet keyword. <i>Valid Values:</i> An ordinal number (starting with 1) of any media defined in the media set determined by the MediaSet keyword.
Prerequisite=(Marks the beginning of the prerequisite description. This type of block can appear multiple times. <i>Valid Values:</i> No values associated with this keyword.
FeatureID=PrereqFeature	Feature ID of a prerequisite object. <i>Valid Values:</i> Any previously defined feature ID.

7.17 Remarks on Netware Requester Installation

There is no response file for the NetWare Requester installation. NetWare requester is installed using two programs: NPNWINST.EXE to install the code and MPTS.EXE to update the PROTOCOL.INI in order to support the IPX protocol.

The NPNWINST.EXE program uses installation parameters. On the BASE.CMD sample file, these parameters are defined by the following variables:

```
nwtgtpath = "C"
nwprefsrv = "NWSRV"
nwtokenring = "TRUE"
nwcontext = ' "'
nwversion = "4"
```

A sample of the update of MPTS is shown in the \CID\RSP\NWMPTS.RSP file on the enclosed CD-ROM.

7.18 Communications Server and Advanced Features Response Files

eNetwork Communications Server for OS/2 Warp and the advanced features use the same installation program, CMSETUP, and the same response file. Keywords are for both Installation (also removal) and configuration.

Note that keywords are added or slightly changed over time. So, response files are version dependent.

To enhance the administrator productivity, these products provide a utility, `CMRECORD`, that generates response files from an existing configuration.

With `CMRECORD`, all features of an existing installation are captured, except the keyboard profiles. If you want to distribute specific keyboard profiles, check the `RESPONSE.INF` in the keyboard record section. Customized keyboard records can be installed with the use of a source configuration specified in the keyboard section.

To avoid typing errors, use `CMSETUP` and `CMRECORD` to create the response files.

1. Create a model configuration using `CMSETUP`. Panels can differ slightly from one version to another, but general rules to follow are:
 1. Enter a new configuration name and a description for the configuration to create.
 2. Answer No to the question whether the configuration will be used for this workstation.
 3. Configure the features you want to install with this model.
 4. Verification is done after selection of Close.
 5. If there are no error messages, exit `CMSETUP`. If you get error messages, do the necessary corrections now.
2. On the same machine, use `CMRECORD` to create a response file.

CMRECORD Basic Syntax

For the complete syntax, see the eNetwork Communications Server for OS.2 Warp *Command Reference* that comes with the product.

```
CMRECORD MYFILE.CFG /O CONFIG.RSP
```

where `MYFILE.CFG` is the name of the configuration file to record, and `CONFIG.RSP` is the name of the output response file.

If a fully qualified path and name is not given for either your configuration or output file, they will be read or write in the current directory.

3. Edit the output response file from `CMRECORD`.

The response file begins by the following lines :

```
*****
* Keywords and values are documented in:
* Response File Reference (Online book).
*
* To use this response file to install or configure...
* using the CMSETUP /R command, values for one or more of these
```

```

* keywords must be provided:
* CMUPDATETYPE =
* CMUSERCFG =
* CMMODELCFG =
* CMSTOPCOMMUNICATIONS =
* CMINSTALLCURRENTFEATURES =
*****

```

where:

CMUPDATETYPE	At least CMUPDATETYPE has to be filled in order to indicate the type of installation. Valid values are: 0 No installation performed 1 New installation or complete reinstallation 2 Upgrade from a previous installation 3 New configuration to use on this workstation 4 New configuration to use on another workstation 5 Force reinstallation of all features 6 Remove code not needed by configuration 7 Install entire code
CMUSERCFG	Specifies the name of the configuration file to be created or updated. If not specified, a default name (ACSCFG) will be used.
CMMODELCFG	Specifies the name of the model configuration to use if any.
CMSTOPCOMMUNICATIONS	Stops communications if it is running and if value specified is 1 or Y.
CMINSTALLCURRENTFEATURES	Valid for a reinstall or update; reinstalls the installed features if value is 1 or 2.

7.18.1 Use of Client-Specific Response File

It is rare that client-specific information is not needed. Plan to use client-specific response files.

Consider how you can group users with similar requirements and provide one default model for each group. There is a variety of options on how to use client-specific response files:

- You can provide a complete response file for each client. This is not such a good idea if you have lots of users and at some time might want to do a global change for all of them, like adding another 5250 session.

- You can use the `INCLUDE` keyword to include a default response file, and then further down in the client's response file, set the keywords you want to be specific.
- You can use the `CMModelCFG` keyword and provide a model configuration to be used and then provide the keywords for the client-specific settings.

INCLUDE Keyword in Response File

When using an `INCLUDE` in a response file, the processing of the keywords in the included file is done immediately. This matches the behavior of the `IncludeInLine` keyword for OS/2.

If a qualified path is not specified, the search order of the include files is:

- The path specified in the `/G:` parameter. This parameter is given with the invocation of `CMSETUP`.
- Current path.
- `DPATH` environment variable.

7.19 eNetwork Personal Communications Response File

There is no utility to create a eNetwork Personal Communications for OS/2 Warp response file. A large number of sample response file is located in the `\RSP` directory of the product CD-ROM.

Keywords are explained in the online documentation. If the product is not installed, it is possible to decompress the documentation files.

Assuming that you want to decompress the files in `D:\INFO`, get into the directory where `DECOMP.EXE` can be found. Then run the following command :

```
DECOMP INFO\PCSREF.IN_ D:\INFO\PCSREF.INF
```

Go into the `D:\INFO` directory and type:

```
VIEW PCSREF.INF
```

Following is an example of a response file for two 3270 sessions with a LUA link to communication server or advanced features.

```
* Emul 3270
EmulType=1
* Session A
Session=(
```

```

        WSName=LUA.WS
    )
Communication=(
    WSName=LUA.WS
    AutoConnect=Y
    Link=rui
    Session=3270
)
3270=(
    WSName=LUA.WS
    ScreenSize=24x80
    SessionType=Display
    HostGraphics=N
    NativeGraphics=N
    LoadPSS=N
    QueryReplyMode=Auto
    CharacterCellSize=9x16
    HostCodePage=297-F
    LtNumber=FF
    LuNumber=2
)
RUI=(
    WSName=LUA.WS
    LUName=LUA
    StartCM=Y
    StopCM=N
    AutoReconnect=Y
)
* Session B
Session=(
    WSName=LUB.WS
)
Communication=(
    WSName=LUB.WS
    AutoConnect=Y
    Link=rui
    Session=3270
)
3270=(
    WSName=LUB.WS
    ScreenSize=24x80
    SessionType=Display
    HostGraphics=N
    NativeGraphics=N
    LoadPSS=N
    QueryReplyMode=Auto
    CharacterCellSize=9x16

```

```

HostCodePage=297-F
LtNumber=FF
LuNumber=3
)
RUI=(
WSName=LUB.WS
LUName=LUB
StartCM=Y
StopCM=N
AutoReconnect=Y
)

```

In this example:

EMULTYPE	Is set to 1 to indicate that personal communication will run only 3270 sessions.
SESSION	Specifies the name of the session. This name is the name of the .WS file that will be found in the PRIVATE subdirectory.
COMMUNICATION	Provides information on the type of link to the host and the type of session. RUI (Request Unit Interface) means LU0/1/2 through communication server or advanced features.
3270	Specifies the characteristics of the 3270 session, in particular LUNumber gives the number of the LU in the PU.
RUI	Defines the parameters required to establish the session.

It is a good idea to also distribute Personal Communications Workstation files (.WS). The following figure is an example of an TCP/IP-based connection:

```

[Profile]
ID=WS
Description=AUSVMR

[Translation]
IBMDefaultView=Y
DefaultView=
IBMDefaultDBCS=Y
DefaultDBCS=

[Communication]
AutoConnect=Y
Link=telnet3270
Session=3270

[SLAN]

```

Adapter=0
DestinationSAP=04
GatewayAddress=400000000000
Identifier=06100000
LinkStations=1
ReceiveBufferCount=12
ReceiveBufferSize=265
SourceSAP=04
TraceName=snalantrace0
TransmitFrameSize=265
CPName=
ExchangeID=N

[LU]
CompEnabled=N
CompBufSize=4096
SLE=N

[Telnet3270]
HostName=ausvmr.austin.ibm.com
HostPortNumber=23
LUName=
AutoReconnect=Y
ExtendedColor=Y
ATTN=6CFF00000000
SYSREQ=F0FF00

[3270]
ScreenSize=32x80
SessionType=Display
HostGraphics=Y
NativeGraphics=N
LoadPSS=N
QueryReplyMode=Auto
CharacterCellSize=9x16
HostCodePage=037-U
LtNumber=FF
LuNumber=FF

[Keyboard]
TypeAhead=Y
ResetInsertbyAttn=N
SpotConversion=Y
CaretSizeControl=Y
CheckCodepage=Y
Language=United-States
IBMDefaultKeyboard=N

DefaultKeyboard=C:\tcpip\pcomos2\PRIVATE\CM101_3.KMP

```
[Window]
SessFlags=8C6A
ViewFlags=4E00
CaptionFormat=2A -
UserTitle=PCSWS
RuleLinePos=0 0
InsertCursor=Y
MFIColor=N
```

```
[Graphics]
Redraw=Retained
CursorStyle=1
CursorMovement=0
```

The following figure is an workstation file example of an SNA-based connection to a 3270 host:

```
[Profile]
ID=WS
Description=VTAM
```

```
[Translation]
IBMDefaultView=Y
DefaultView=
IBMDefaultDBCS=Y
DefaultDBCS=
```

```
[Communication]
AutoConnect=Y
Link=slan
Session=3270
```

```
[SLAN]
Adapter=0
DestinationSAP=04
GatewayAddress=400021031054
Identifier=05D5AB54
LinkStations=1
ReceiveBufferCount=12
ReceiveBufferSize=2012
SourceSAP=04
TraceName=snalantrace0
TransmitFrameSize=2012
CPName=
```

```
ExchangeID=N

[LU]
CompEnabled=N
CompBufSize=4096
SLE=N

[Telnet3270]
HostName=wtscpok.itso.ibm.com
HostPortNumber=23
LUName=
AutoReconnect=Y
ExtendedColor=Y
ATTN=6CFFFEFF3
SYSREQ=F0FFEF

[3270]
ScreenSize=32x80
SessionType=Display
HostGraphics=Y
NativeGraphics=N
LoadPSS=N
QueryReplyMode=Auto
CharacterCellSize=9x16
HostCodePage=037-U
LtNumber=FF
LuNumber=FF

[Keyboard]
TypeAhead=Y
ResetInsertbyAttn=N
SpotConversion=Y
CaretSizeControl=Y
CheckCodepage=Y
Language=United-States
IBMDefaultKeyboard=N
DefaultKeyboard=C:\tcpip\pcomos2\PRIVATE\CM101_3.KMP
```

7.20 Lotus Notes OS/2 Client Response File

There is no utility to create a Lotus Notes OS/2 client response file. A sample response file RESPONSE.RSP is located on the product's CD-ROM in the \OS2\INSTALL directory.

Lotus products use the Software Installer to install the product either locally or remotely. The following figure displays a response file that installs the Lotus Notes client.

```
FILE=D:\NOTES
WORK=D:\NOTES\DATA

COMP=Installer
COMP=Notes Workstation
* COMP=Domino Server
COMP=Notes Data
COMP=Notes Viewers
COMP=Notes Local Required Data
COMP=Notes Required Data
COMP=Personal data files
COMP=Documentation Databases
* COMP=Notes Release 4 Help
COMP=Notes Help Lite
COMP=Additional Templates
COMP=Additional Dictionaries

CFGUPDATE=AUTO
OVERWRITE=YES
SAVEBACKUP=YES
DELETEDBACKUP=NO
```

Figure 70. Lotus Notes Client Response File

Note: Keywords are case-sensitive.

7.21 Lotus SmartSuite 96 Response File

On the product's CD-ROM, there is a sample SUITE.RSP response file. The following figure shows a response file that installs all features but SmartCenter (OS/2 Warp 4 comes with the Warp Center, which is an enhanced version of Lotus SmartCenter).

```
USER_NAME = ITSC Austin
COMPANY_NAME = IBM Corporation
CFGUPDATE = AUTO
FILE = D:\SSUITE96
AUX1 = D:\SSUITE96
FOLDER_NAME = Lotus SmartSuite
INSTALL_TYPE = 0
INSTALL_OPTION = 1
SM_BACKUP = NO
COMP = Freelance Graphics for OS/2
COMP = Word Pro for OS/2
COMP = 1-2-3 and Freelance File Translator
AUX2 = D:\SSUITE96\FLG
FLG_INSTALL_OPTION = 1
AUX3 = D:\SSUITE96\WORDPRO
WPRO_INSTALL_OPTION = 1
AUX4 = D:\SSUITE96\TRANS
OVERWRITE = YES
SAVEBACKUP = NO
DELETEDBACKUP = NO
```

Figure 71. Lotus SmartSuite 96 Response File

To get to know all possible keywords, edit the SUITE.RSP file. It is possible to customize the install in great detail.

Part 2. Installing and Using Software Distribution

Chapter 8. Optimizing OS/2 Warp 4 Boot Diskettes

This chapter provides information how to get the maximum out of boot diskettes. We show how to provide support for TCP/IP, TCPBEUI, and NFS on the set of three boot diskettes.

8.1 Creating OS/2 Boot Diskettes

To create boot diskettes, you basically need to perform the following steps:

1. Create OS/2 base operating boot diskettes.
2. Install network adapter/network protocol support.
3. Install a redirect and/or software distribution agent.

These steps are nearly independent of the software distribution program you are going to use. There are, however, dependencies between the individual steps.

Note: If you are using NVDM/2, the redirector is added when you install the software distribution support. So you don't have to install the redirector; you only need the OS/2 base boot diskettes, the network support and the software distribution support.

Normally, you can use the boot diskettes for different workstations. Every client needs a different "name" as identification. Most of the redirectors allow prompting for the workstation name. Alternatively, you can customize the workstation name directly in the setup files on the boot diskettes. If you are using workstations with different hardware configurations, you should set up a set of boot diskettes for every type of workstation you are going to install.

To create the base OS/2 boot diskettes, we recommend using `SEDISK` as illustrated in "The `SEDISK` Command" on page 79.

The following figure shows the files needed on the three boot diskettes of OS/2 Warp 4 to run a base OS/2 operating system. They are examples of how to create base boot diskettes with enough space to add additional files.

To get sufficient free space, we have packed some of the files we do not need at boot time in ZIP files. In addition, we install a virtual diskette with about 2 MB of disk space. The ZIP file is unpacked to the virtual diskette drive, and the path statements are adjusted to reflect the new location.

If you are deleting files from the original OS/2 boot diskettes, you have to update the CONFIG.SYS file and the SNOOP.LST file on Diskette 1. If you delete files without updating these files, you will receive error messages. If you create your boot diskettes with SEINST or CDINST, you will find more files on the diskettes:

8.1.1 Contents of Diskette 0 (OS/2 Warp 4 Installation Diskette)

The following files should reside on the very first diskette of the set of three boot diskettes.

OS2BOOT	000000.BIO	ABIOS.SYS	ALTF1.COMD	ALTF1BOT.SCR
ALTF1TOP.SCR	CONFIG.X	F80000.BIO	F80100.BIO	80200.BIO
F80402.BIO	F80403.BIO	F80404.BIO	F80600.BIO	80700.BIO
F80701.BIO	F80702.BIO	F80703.BIO	F80704.BIO	F80902.BIO
F80903.BIO	F80904.BIO	F80A00.BIO	F80A01.BIO	F80A02.BIO
F80C00.BIO	F80D00.BIO	F80D01.BIO	F81000.BIO	F81B00.BIO
F88000.BIO	FC0400.BIO	FC0403.BIO	FC0500.BIO	OS2DUMP
OS2KRNL1	OS2LDR	OS2LDR.MSG	OS2VER	SYSINSTX.COM
W020100.BIO	W020101.BIO	W050000.BIO	W050100.BIO	W050101.BIO
W060100.BIO	W0F0000.BIO	XDF.MSG	XDFCOPY.EXE	XDFH.MSG

The first diskette is unchanged because you cannot put any other files used at boot time on this diskette.

8.1.2 Contents of Diskette 1 (OS/2 Warp 4 Diskette 1)

The following files reside on the second diskette of the set of three boot diskettes:

AHA152X.ADD	AHA154X.ADD	AHA154X.SNP	AHA164X.ADD	AHA174X.ADD
AHA174X.SNP	AHA6360.SNP	AIC7770.ADD	AIC7770.SNP	AIC7870.ADD
AIC7870.SNP	BTSCSI.ADD	BTSCSI.SNP	CLOCK01.SYS	CLOCK02.SYS
CONFIG.SYS	CONFIG.X	DAC960.ADD	DAC960.SNP	DPT20XX.ADD
DPT20XX.SNP	FD16-700.ADD	FD16-700.SNP	FD7000EX.ADD	FD7000EX.SNP
FD8XX.ADD	FD8XX.SNP	FLASHPT.ADD	FLASHPT.SNP	IBM1FLPY.ADD
IBM1FLPY.SNP	IBMS506.ADD	IBMS506.SNP	IBM2ADSK.ADD	IBM2FLPY.ADD
IBM2SCSI.ADD	IBMIDEC.DFLT	IBMINT13.I13	IBMKBD.SNP	IBMKBD.SYS
IPSRAID.ADD	IPSRAID.SNP	IR.SNP	ISAPNP.SNP	KBDBASE.SYS
NETDET1.SNP	NETDET2.SNP	OS2DASD.DMD	OS2LOGO	PARALLEL.SNP
PCIBUS.SNP	PCMCIA.SNP	PNP.SYS	QL100S2.ADD	QL100S2.SNP
QL400S2.ADD	QL400S2.SNP	QL510.ADD	QL510.SNP	RESERVE.SYS
RESOURCE.SYS	RESRV.SNP	SCREEN01.SYS	SCREEN02.SYS	SERIAL.SNP
SNOOP.LST	SONY535.ADD	TMV1SCSI.ADD	XDFLOPPY.FLT	

8.1.2.1 The CONFIG.SYS File

The following example displays the CONFIG.SYS file.

```

buffers=32
ioopl=yes
menman=swap,delayswap
protshell=cmd.exe /K a:\startup.cmd
set os2_shell=cmd.exe
diskcache=D2,LW
protectonly=yes
libpath=.\;
ifs=hpfs.ifs /c:64
pauseonerror=no
codepage=850
devinfo=kbd,us,keyboard.dcp
devinfo=scr,ega,vt1850.dcp
device=\dos.sys
set path=.\;x:\cmd;x:\exe;x:\dll;
set dpath=.\;x:\cmd;x:\exe;
set keys=on
basedev=ibmkbd.sys
basedev=ibm1flpy.add
basedev=ibms506.add
basedev=ibm2flpy.add
basedev=ibm2adsk.add
basedev=ibm2scsi.add
basedev=ibmint13.i13
basedev=os2dasd.dmd
device=\testcfg.sys
basedev=xdfloppy.flt

```

```
set saveconnect=1
basedev=aha152x.add
basedev=aha154x.add
basedev=aha164x.add
basedev=aha174x.add
basedev=aic7770.add
basedev=aic7870.add
basedev=btscsi.add
basedev=fd16-700.add
basedev=fd8xx.add
basedev=fd7000ex.add
basedev=dpt20xx.add
basedev=dac960.add
basedev=flashpt.add
basedev=ipsraid.add
basedev=ql10os2.add
basedev=ql40os2.add
basedev=ql510.add
device=\vdisk.sys 2000,,
```

8.1.2.2 The SNOOP.LST File

The following example displays the SNOOP.LST file.

```
resrv.snp
netdet1.snp
ibmkbd.snp
ibm1flpy.snp
ibm1s506.snp
; SCSI Snoopers
aha6360.snp
aha154x.snp
aha174x.snp
aic7870.snp
ql10os2.snp
ql40os2.snp
ql510.snp
ipsraid.snp
btscsi.snp
fd16-700.snp
fd8xx.snp
fd7000ex.snp
dpt20xx.snp
flashpt.snp
dac960.snp
; Misc. Snoopers
pcmcia.snp
ir.snp
```

```

netdet2.snp
; CDRom Snoopers
; Audio Snoopers
; Misc. Snoopers
parallel.snp
serial.snp
; Note: Place additional snoopers above this line
pcibus.snp

```

If you need more disk space on Diskette 1 or Diskette 2 for additional device drivers, you can delete some of the SCSI drivers you do not need for your target machine. Files for additional device drivers that are included with BASEDEV statement must be on Diskette 1. Drivers included with a DEVICE statement must be added on Diskette 2.

8.1.2.3 Contents of Diskette 2

The following example displays all files that reside on the third diskette of the set of three boot diskettes.

BKSCALLS.DLL	BMSCALLS.DLL	BVHINIT.DLL	BVSCALLS.DLL	CMD.EXE
COUNTRY.SYS	DOS.SYS	DOSCALL1.DLL	FILES.ZIP	HPFS.IFS
KBDCALLS.DLL	KEYBOARD.DCP	MSG.DLL	NAMPIPES.DLL	NLS.DLL
OS2CHAR.DLL	OSO001.MSG	PKUNZIP2.EXE	QUECALLS.DLL	SESMGR.DLL
STARTUP.COM	TESTCFG.SYS	VDISK.SYS	VIOCALLS.DLL	VTBL850.DCP

These files are needed to boot the OS/2 System, create the virtual disk and unpack the ZIP file. If you need more space on the boot disk, you can delete the OSO001.MSG file, add it to the FILES.ZIP file or move it to a redirected drive. You will receive a error message at boot time because VDISK.SYS wants to access OSO001.MSG to display a status message, but you can ignore this error message.

All other files, like UNPACK.EXE, UNPACK2.EXE, FORMAT.COM, and CHKDSK.COM, are located on the redirected drive and can be accessed after the redirector is running.

8.1.2.4 Contents of FILES.ZIP

The following example displays the files that are zipped to FILES.ZIP.

The STARTUP.COM is called directly form the CONFIG.SYS to detect the virtual diskette drive, copy files form Diskette 2 to the virtual diskette and unpack the ZIP file(s).

Archive: FILES.ZIP							
Length	Method	Size	Ratio	Date	Time	CRC-32	Name
-----	-----	-----	-----	-----	-----	-----	-----
9390	Implode	8756	7%	08-09-96	01:02	70930ff7	TEDIT.EXE
212	Shrunk	179	16%	06-14-96	10:58	89d1ed7a	TEDIT.HLP
21812	Implode	17366	20%	08-12-96	03:21	9b48f193	NPXEMPLTR.DLL
512	Implode	233	55%	08-12-96	03:00	51beb28b	ANSICALL.DLL
9415	Implode	8180	13%	08-13-96	11:05	16f40fa6	HARDERR.EXE
9086	Implode	2770	70%	08-12-96	01:42	f340ba20	MARKETING.MSG
-----	-----	-----	-----	-----	-----	-----	-----
50427		37484	26%				6

The STARTUP.CMD also adjusts the path statements. The drive letter of the virtual diskette is added in front of the previous statements. This will cause all programs to look first on the virtual diskette before looking for Diskette 2.

Note on VDISK.SYS

If you have many files in your ZIP files, or if you want to copy the complete Diskette 2 to the virtual diskett drive you may have to adjust the VDISK.SYS statment in the CONFIG.SYS

```
VDISK.SYS <kilobytes>, <sectors>, <directories>
```

Increase the size of <directories> for more directory entries.

8.1.2.5 Contents of the STARTUP.CMD File

The following example displays the contents of the STARTUP.CMD file.

```
@echo off
set vdisk=k:
copy a:*.zip %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=i:
copy a:*.zip %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=h:
copy a:*.zip %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=g:
copy a:*.zip %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=f:
copy a:*.zip %vdisk% 1>nul 2>nul
```

```

if not errorlevel 1 goto config:
set vdisk=e:
copy a:*.zip %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
set vdisk=d:
copy a:*.zip %vdisk% 1>nul 2>nul
if not errorlevel 1 goto config:
goto error
:config
%vdisk%
a:\pkunzip2 -d *.zip %vdisk%
copy a:\*.dll %vdisk%
copy a:\*.exe %vdisk%
set path=%vdisk%\;%path%
set dpath=%vdisk%\;%dpath%
set beginlibpath=%vdisk%\;a:\;
copy a:\*.cmd %vdisk%
if not exist a:\setup.cmd goto end:
copy a:\setup.cmd %vdisk%\setup.cmd
%vdisk%\setup.cmd
goto end:
:error
Echo Error finding VDISK
:end

```

The STARTUP.CMD is also looking for a SETUP.CMD. If it is found, it is executed. The SETUP.CMD is used to start/configure the network connection and starts the CID installation. All the changes you need for your environment should be made in the SETUP.CMD file.

8.2 Adding Network Adapter Support

The easiest way to add the network adapter support is to execute the `THINLAPS` (Chapter 4.11.1, “The THINLAPS Command” on page 112) command. You can choose the network adapter you have installed in your workstation, and the CONFIG.SYS on Diskette 1 is customized.

The disadvantage of this method is that you have files on your diskettes that are not actually needed. Especially if you are using TCPBEUI, you will recognize that you have space problems.

In this example, the `THINLAPS` command is used, but after the installation, some of the files are deleted. Other files are added to a ZIP file, and the configuration is also customized.

At the time you are adding the network adapter support, you also define the network protocol to use. With THINLAPS you can install:

- NetBIOS
- TCP/IP
- TCPBEUI

8.2.1 NetBIOS Support

To install NetBIOS support to your boot diskettes, use the following command:

NetBIOS Support

```
THINLAPS D:\CID\IMG\MPTS A: IBMTOK.NIF
```

THINLAPS.EXE copies the network adapter and the network protocol files to Diskette 2:

LT2.MSG	IBMTOK.OS2	PROTMAN.OS2	PRO.MSG	LT0.MSG
LANMSGDD.OS2	LANMSGEX.EXE	LANMSGDL.DLL	ACSNETB.DLL	NETBEUI.OS2
NETBIOS.OS2	NETBIND.EXE	PROTOCOL.INI		

The CONFIG.SYS file on Diskette 1 is updated with:

```
rem *** Start of ThinLAPS additions ***
call = netbind.exe
run = lanmsgex.exe
device = lanmsgdd.os2
device = protman.os2 /I:a:\
device = netbeui.os2
device = netbios.os2
device = IBMTOK.OS2
rem *** End of ThinLAPS additions ***
```

With NetBIOS, there should be no space problems; so we did not create a ZIP file for it.

8.2.2 TCP/IP Support

To install TCP/IP support to your boot diskettes, use the following single-line command (while adjusting network parameters to your environment):

TCPIP Support

```
THINLAPS D:\CID\IMG\MPTS A: IBMTOK.NIF /TCPIP /IP:9.3.1.97 /RT:9.3.1.74  
/NM:255.255.255.0 /DMN:itsc.austin.ibm.com
```

THINLAPS.EXE copies the network adapter and the network protocol files to Diskette 2:

LT2.MSG	IBMTOK.OS2	PROTMAN.OS2	PRO.MSG	LT0.MSG
LANMSGDD.OS2	LANMSGEX.EXE	LANMSGDL.DLL	[ETC]	ARP.EXE
CNTRL.EXE	IFCONFIG.EXE	INETWAIT.EXE	ROUTE.EXE	AFINET.SYS
IFNDIS.SYS	SOCKETS.SYS	SO32DLL.DLL	TCP32DLL.DLL	TCPIDLL.DLL
TCPMRI.DLL	NETBIND.EXE	PROTOCOL.INI	MPTSTART.CMD	SETUP.CMD

The file called RESOLV2 must be copied manually:

[.]	[..]	DHCPD.CFG	PROTOCOL	SERVICES
RESOLV2				

On Diskette 1, the CONFIG.SYS is updated with:

```
rem *** Start of ThinLAPS additions ***  
call = netbind.exe  
run = lanmsgex.exe  
device = lanmsgdd.os2  
device = protman.os2 /I:a:\  
device = sockets.sys  
device = afinet.sys  
device = ifndis.sys  
run = cntrl.exe  
call = cmd.exe /Q /C a:\mptstart.cmd  
device = IBMTOK.OS2  
set etc=a:\etc  
rem *** End of ThinLAPS additions ***
```

To configure TCP/IP, THINLAPS creates two files on Diskette 2:

- MPTSTART.CMD

```
@ECHO OFF  
echo MPTS initialization is starting.
```

```
IF NOT EXIST a:\SETUP.CMD GOTO END
INETWAIT
CALL a:\SETUP.CMD
:END
echo MPTS initialization is complete.
```

- **SETUP.CMD**

```
route -fh
arp -f
ifconfig lan0 9.3.1.97 netmask 255.255.255.0 metric 0 mtu 1500
route add net 9 9.3.1.74 1
route add default 9.3.1.74 1
```

To avoid problems with space on the boot diskettes, you should customize Diskette 1 and Diskette 2.

This is an example of how you can customize the TCP/IP configuration. All the files needed for TCP/IP configuration are stored in the TCPIP.ZIP file on Diskette 2. Only the device drivers and related files are in uncompressed form on Diskette 2.

8.2.2.1 Contents of Diskette 2

The following example displays the files that reside on Diskette 2.

AFINET.SYS	IBMTOK.OS2	IFNDIS.SYS	LANMSGDD.OS2	LANMSGDL.DLL
LANMSGEX.EXE	LT0.MSG	LT2.MSG	NETBIND.EXE	PRO.MSG
PROTMAN.OS2	PROTOCOL.INI	SETUP.CMD	SOCKETS.SYS	TCPIP.ZIP

8.2.2.2 Contents of TCPIP.ZIP

The following example displays the files zipped together in TCPIP.ZIP.

Archive: TCPIP.ZIP							
Length	Method	Size	Ratio	Date	Time	CRC-32	Name
-----	-----	-----	-----	-----	-----	-----	-----
19893	Implode	12229	39%	08-23-96	13:49	538f6181	ARP.EXE
11587	Implode	6549	44%	08-23-96	13:49	a489c606	CNTRL.EXE
17193	Implode	10523	39%	08-23-96	13:49	08bae5fe	IFCONFIG.EXE
24064	Implode	13573	44%	08-23-96	13:49	39b7c9fd	INETWAIT.EXE
20671	Implode	12381	40%	08-23-96	13:49	e24f13c8	ROUTE.EXE
37842	Implode	17144	55%	08-23-96	13:49	c9d04c02	SO32DLL.DLL
62289	Implode	31557	49%	08-23-96	13:49	bcd3a206	TCP32DLL.DLL
47279	Implode	24847	47%	08-23-96	13:49	b3ebb09d	TCPIPDLL.DLL
34304	Implode	10813	69%	08-11-96	22:19	0a1976b7	TCPMRI.DLL
13218	Implode	3838	71%	08-11-96	22:05	205351ef	ETC/DHCPD.CFG
5451	Implode	2344	57%	08-11-96	22:05	9923695a	ETC/PROTOCOL
29	Stored	29	0%	08-19-97	11:25	207339db	ETC/RESOLV2
42725	Implode	10108	76%	08-11-96	22:05	3c529371	ETC/SERVICES
-----	-----	-----	-----	-----	-----	-----	-----
336545		155935	54%				13

8.2.2.3 Contents of SETUP.CMD

The following example displays the contents of the SETUP.CMD file.

```
set etc=%vdisk%\etc
detach cntrl.exe
inetwait
route -fh
arp -f
ifconfig lan0 9.3.1.97 netmask 255.255.255.0 metric 0 mtu 1500
route add default 9.3.1.74 1
```

8.2.2.4 Added Lines to CONFIG.SYS on Diskette 1

The following example displays the lines that were added to CONFIG.SYS.

```
Rem Additions for TCPIP
call = netbind.exe
run = lanmsgex.exe
device = lanmsgdd.os2
device = protman.os2 /I:a:\
device = sockets.sys
device = afinet.sys
device = ifndis.sys
device = IBMTOK.OS2
```

Note

The AFINET.SYS can be replaced with the AFLEAN.SYS, which is a smaller version, saving approximately 100 KB. The AFLEAN.SYS should be available as APAR IP20945 in the first OS/2 Warp 4 MPTS correctivte services package (WR08415)..

8.2.3 Installing TCPBEUI Support

To install TCPBEUI support to your boot diskettes, use the following single-line command (while adjusting the network parameters to your environment):

TCPBEUI Support

```
THINLAPS D:\CID\IMG\MPTS A: IBMTOK.NIF /TCPBEUI /IP:9.3.1.97
/RT:9.3.1.74 /NM:255.255.255.0 /DMN:itsc.austin.ibm.com
```

THINLAPS copies the network adapter and the network protocol files to Diskette 2:

LT2.MSG	IBMTOK.OS2	PROTMAN.OS2	PRO.MSG	LT0.MSG
LANMSGDD.OS2	LANMSGEX.EXE	LANMSGDL.DLL	ACSNETB.DLL	TCPBEUI.OS2
NETBIOS.OS2	[ETC]	ARP.EXE	CNTRL.EXE	IFCONFIG.EXE
INETWAIT.EXE	ROUTE.EXE	AFINET.SYS	IFNDIS.SYS	SOCKETS.SYS
SO32DLL.DLL	TCP32DLL.DLL	TCPIPDLL.DLL	TCPMRI.DLL	NBTCP.EXE
NETBIND.EXE	PROTOCOL.INI	MPTSTART.CMD	SETUP.CMD	

Make sure RESOLV2 has been copied, too:

[.]	[..]	DHCPD.CFG	PROTOCOL	SERVICES
RESOLV2				

In addition, you need to copy the file TCPTIME.DLL to Diskette 2. This file usually resides in the \MPTN\DLL directory of an OS/2 Warp 4 system with TCP/IP installed.

On Diskette 1, the CONFIG.SYS file is updated with the following lines:

```
rem *** Start of ThinLAPS additions ***
call = netbind.exe
run = lanmsgex.exe
```

```

device = lanmsgdd.os2
device = protman.os2 /I:a:\
device = sockets.sys
device = afinet.sys
device = ifndis.sys
run = cntrl.exe
call = cmd.exe /Q /C a:\mptstart.cmd
device = tcpbeui.os2
device = netbios.os2
device = IBMTOK.OS2
run = nbtcp.exe
set etc=a:\etc
rem *** End of ThinLAPS additions ***

```

To configure TCP/IP, THINLAPS creates two files on Diskette 2:

- **MPTSTART.CMD**

```

@ECHO OFF
echo MPTS initialization is starting.

IF NOT EXIST a:\SETUP.CMD GOTO END
INETWAIT
CALL a:\SETUP.CMD
:END
echo MPTS initialization is complete.

```

- **SETUP.CMD**

```

route -fh
arp -f
ifconfig lan0 9.3.1.97 netmask 255.255.255.0 metric 0 mtu 1500
route add net 9 9.3.1.74 1
route add default 9.3.1.74 1

```

You will recognize that there is not enough space on Diskette 2 to copy all the files to disk. The workaround is to create the THINLAPS configuration on an empty disk, customize it and add it then to Diskette 2.

8.2.3.1 Customized TCPBEUI Boot Diskettes

Here is the example of a customized version of TCPBEUI:

Contents of Diskette 2

The following example displays the contents of boot Diskette 2.

AFINET.SYS	IBMTOK.OS2	IFNDIS.SYS	LANMSGDD.OS2	LANMSGDL.DLL
LANMSGEX.EXE	LT0.MSG	LT2.MSG	NETBIND.EXE	NETBIOS.OS2
PRO.MSG	PROTMAN.OS2	PROTOCOL.INI	SETUP.CMD	SOCKETS.SYS
TCPBEUI.OS2	TCPBEUI.ZIP			

Contents of the TCPBEUI.ZIP File

The following example displays the contents of the TCPBEUI.ZIP file.

```

Archive:  TCPBEUI.ZIP
Length  Method  Size  Ratio  Date    Time    CRC-32  Name
-----  -
 3349  Implode  2107  37%   08-01-96  18:05  68a1f022  ACSNETB.DLL
19893  Implode 12229  39%   08-23-96  13:49  538f6181  ARP.EXE
11587  Implode  6549  44%   08-23-96  13:49  a489c606  CNTRL.EXE
17193  Implode 10523  39%   08-23-96  13:49  08bae5fe  IFCONFIG.EXE
24064  Implode 13573  44%   08-23-96  13:49  39b7c9fd  INETWAIT.EXE
40960  Implode 21293  48%   08-11-96  22:21  118b7eae  NBTCP.EXE
   75  Shrunk   45    40%   08-26-97  08:58  4f95f0a7  rfcNAMES.LST
20671  Implode 12381  40%   08-23-96  13:49  e24f13c8  ROUTE.EXE
37842  Implode 17144  55%   08-23-96  13:49  c9d04c02  SO32DLL.DLL
62289  Implode 31557  49%   08-23-96  13:49  bcd3a206  TCP32DLL.DLL
47279  Implode 24847  47%   08-23-96  13:49  b3ebb09d  TCPIPDLL.DLL
34304  Implode 10813  69%   08-11-96  22:19  0a1976b7  TCPMRI.DLL
32288  Implode 18076  44%   08-23-96  13:49  32cbea9e  TCPTIME.DLL
13218  Implode  3838  71%   08-11-96  22:05  205351ef  etc/DHCPD.CFG
 2780  Implode  1222  56%   08-20-97  15:27  942eb7e9  etc/PROTOCOL
   29  Stored    29    0%   08-19-97  11:25  207339db  etc/RESOLV2
21058  Implode  5473  74%   08-20-97  15:33  74138b31  etc/SERVICES
-----  -
388879          191699  51%                   17

```

Added Lines to CONFIG.SYS

The following example displays the lines added to the CONFIG.SYS file.

```

rem *** Start of ThinLAPS additions ***
call = netbind.exe
run = lanmsgex.exe
device = lanmsgdd.os2
device = protman.os2 /I:a:\
device = sockets.sys
device = afinet.sys
device = ifndis.sys
device = tcpbeui.os2

```

```
device = netbios.os2
device = IBMTOK.OS2
rem *** End of ThinLAPS additions ***
```

Contents of SETUP.CMD

The following example displays the contents of the SETUP.CMD file.

```
set etc=%vdisk%\etc
set path=%path%x:\dll\os2;x:\locintsu;x:\exe;
set dpath=%dpath%x:\dll\os2;x:\locinstu;x:\exe;
set endlibpath=x:\dll\os2;x:\locinstu;
detach cntrl.exe
inetwait
arp -f
route -fh
ifconfig lan0 9.3.1.97 netmask 255.255.255.0
route add default 9.3.1.74 1
detach nbtcp.exe
```

In our environment, we have a NetBIOS name server running. If you are not using a NetBIOS name server, you should maintain the RFCNAMES.LST file to ensure that the NetBIOS name of your CID server can be resolved to a valid TCP/IP address.

8.3 Adding a Redirector

Depending on the the network protocol you use in your environment, you have to choose one of the three following redirectors:

- ANXIFS
Can be used with NetBIOS and TCPBEUI
- SRVIFS
Can be used with NetBIOS and TCPBEUI
- NFS
Can be used only with TCP/IP

8.3.1 ANXIFS

The ANXIFS is normally used with the TME 10 Software Distribution product. It is also used with NVDM/2, but it is installed with the `NVDMBDSK` command (See “NVDM/2” on page 306).

The files for ANXIFS can be found in the file ANXIFS.ZIP file in the `<x:>\SD4OS2\PRISTINE` directory. You should unpack the file to `\CID\IMG\ANXIFS`.

On the boot diskettes, you need:

- ANXCLT.INI
- ANXIFCOM.IFS
- ANXIFCOM.SYS
- ANXIFPID.SYS
- ANXIFS.MSG
- ANXREQ.EXE

In addition, the file OSO001.MSG is used, but if you have problems with disk space, you can omit that file. Copy the files on Diskette 2. For further information, see “Configure Redirection Support” on page 409.

If desired, you can add the files ANXREQ.EXE and ANXCLT.INI to a ZIP file which is automatically unzipped to the virtual diskette drive. To start the redirector, add the `ANXREQ` statement out of `FNDVNB.CMD` to the `SETUP.CMD` file.

8.3.1.1 Redirector-Specific Lines in CONFIG.SYS

The following example displays the redirector-specific device drivers in the `CONFIG.SYS` file.

```
DEVICE=\ANXIFPID.SYS
DEVICE=\ANXIFCOM.SYS
IFS=\ANXIFCOM.IFS
```

8.3.1.2 Added Line to SETUP.CMD

The following line is added to the `SETUP.CMD` file:

```
ANXREQ START @ANXCLT.INI
```

`ANXCLT.INI` controls the redirector and defines the drives to be attached. If you do not specify a client name, a unique computer name is generated by `ANXIFS`. So you can use the diskettes for different workstations without changing the `ANXCLT.INI` file.

8.3.1.3 Contents of ANXCLT.INI

The following example displays the contents of the `ANXCLT.INI` file.

```
;Adapter Number supported 0-1
adapter=0
;Max numbers of attach
numattaches=4
;clientname=ff - no clientname set
;SDSERV1 is the Anxifs Server Name
```

```
;REMAGT - remote Agent
;CODESRV - images, rsp and logfiles
attach=z,SDSERV1,REMAGT
attach=x,SDSERV1,CODESRV
```

8.3.2 SrvIFS

All the files needed for SrvIFS can be installed with the `THINIFS` command as illustrated in "The THINIFS Command" on page 101.

The files needed for the redirector are:

- SRVIFS.SYS
- SRVIFSC.IFS
- SRVATTCH.EXE

The SRVIFS.SYS driver and the SRVIFSC.IFS file must be copied to Diskette 2. The SRVATTCH.EXE file can be added to any of the existing ZIP files. You can create also a new ZIP file. This is an example using SrvIFS with TCPBEUI.

8.3.2.1 Added Lines to CONFIG.SYS

The following example displays the lines added to the CONFIG.SYS file for the SrvIFS redirector.

```
DEVICE=\SRVIFS.SYS
IFS=\SRVIFSC.IFS * /
```

8.3.2.2 Contents of SETUP.CMD

The following example displays the contents of the SETUP.CMD file.

```
set etc=%vdisk%\etc
detach cntrl.exe
inetwait
arp -f
route -fh
ifconfig lan0 9.3.1.97 netmask 255.255.255.0
route add default 9.3.1.74 1
detach nbtcp.exe
srvattch x: \\CIDLCU\cid
srvattch z: \\CIDLCU\log
```

8.3.3 NFS

The NFS files needed to establish a connection in addition to the TCP/IP files are:

- NFS200.IFS

- NFSCTL.EXE
- NFSWAIT.EXE
- MOUNT.EXE
- RPCDLL.DLLI
- TCPTIME.DLL

To save disk space, add some of the files to the TCPIP.ZIP file. You need the NFS200.IFS file at boot time; so don't add this file to the ZIP file. All the other files are not needed at boot time and can be added to the ZIP file.

8.3.3.1 Added Line to CONFIG.SYS

The following line is added to the CONFIG.SYS file to support NFS:

```
IFS=\NFS200.IFS
```

8.3.3.2 Added Lines to SETUP.CMD

The following statements must be added to SETUP.CMD to access the code server directories:

```
nfswait  
detach nfsctl  
mount -u0 -g0 x: 9.3.1.85 d:\cid\img  
mount -u0 -g0 l: 9.3.1.85 d:\cid\log
```

If you use TCP/IP with a NFS connection to the code server, make sure that all necessary files for the NFS connection are installed at the target workstation before the machine reboots.

More NFS Information

Also check Chapter B.6, "Creating OS/2 Warp 4 NFS CID Boot Diskettes" on page 492 for more information on NFS-related issues.

8.4 Adding Software Distribution Agent

In this redbook, we use three different software distribution managers:

- LCU (LAN CID Utility)
Works with NetBIOS and TCPBEUI
- NVMD/2 (NetView Distribution Manager/2)
Works with NetBIOS and TCPBEUI

- Tivoli TME10 SD 3.1.3
Works with TPCIP, NetBIOS and TCPBEUI

Remember that the hard disk must be partitioned before you start the actual installation of OS/2 Warp 4. You can either do that manually, install over an existing version of OS/2, or use the software distribution manager to handle the partitioning and formatting for you.

8.4.1 LCU

The installation of LCU is done by CASINSTL (see “The CASINSTL Command” on page 107). You have no new files on your boot diskettes, but the STARTUP.CMD and the CONFIG.SYS will be updated. In the following examples `<drive:\path>` refers to the redirected drive.

8.4.1.1 Added Line to CONFIG.SYS

The following line is added to the CONFIG.SYS file when CASINSTL is run:

```
RUN=<drive:\path>SRVREXX.EXE
```

8.4.1.2 Contents of STARTUP.CMD

The following figure displays the contents of the STARTUP.CMD file.

```
<drive:\path>CASAGENT.EXE /REQ:*P /CMD<drive:\path>\cmd /D
CMD
EXIT
```

If you are using a customized boot diskettes it is useful to remove the `RUN=X:\LCU\SRVREXX.EXE` statement from the CONFIG.SYS and move it to the STARTUP.CMD. This is done so you can be sure, that the redirector is already up and running and you can access the redirected file system.

8.4.1.3 Alternate STARTUP.CMD

The following STARTUP.CMD can be used to load REXX support off the code server rather than locally to save disk space on the diskettes.

```
set path=%path%x:\dll\os2;x:\locintsu;x:\exe;
set dpath=%dpath%x:\dll\os2;x:\locinstu;x:\exe;
set endlibpath=x:\dll\os2;x:\locinstu;
```

```
detach x:\locinstu\srvrexx.exe
x:\locinstu\casagent.exe /REQ:*P /CMD:X:\CLIENT /D
```

The steps for creating boot diskettes using LCU are documented in “Creating LCU Boot Diskettes” on page 319.

8.4.2 NVDM/2

For detailed information about creating NetView DM/2 boot diskettes, please refer also to "Creation of Boot Diskettes" on page 354. You can use the `NVDMBDSK` command to copy the NVDM client code to the boot diskettes.

Rather than using `NVDMBDSK`, you may to copy the files needed to the diskettes and update the appropriated configuration files. All configuration parameters are stored in the `IBMNVDM2.INI` file.

NVDM/2 uses a different version of ANXIFS that is installed when you use the `NVDMBDSK` command. There is no definition file for the connection to the server machine. The Z: drive is always assigned by default to the remote agent directory and reflects the server's PCACODE directory.

We are using a NetBIOS connection in this example, and the files of NVDM/2 are added to a subdirectory on Diskette 2.

```
\IBMNVDM2
[.]          [..]          [DLL]          [BIN]          IBMNVDM2.INI

\IBMNVDM2\DLL
[.]          [..]          ANXIFREQ.DLL   NDMNLSP.DLL

\IBMNVDM2\BIN
[.]          [..]          ANXPULAG.EXE   ANXACAIP.SYS   ANXIFCOM.SYS
ANXIFCOM.IFS ANXIFPID.SYS
```

If you need more disk space on Diskette 2, you can add the executable to a ZIP file and unpack it into the `\IBMNVDM2\BIN` subdirectory on the virtual diskette drive.

8.4.2.1 Added Lines to CONFIG.SYS

The following lines are added to the `CONFIG.SYS` file to load the NVDM/2 client:

```
DEVICE=\IBMNVDM2\BIN\ANXACAIP.SYS
DEVICE=\IBMNVDM2\BIN\ANXIFPID.SYS
DEVICE=\IBMNVDM2\BIN\ANXIFCOM.SYS
IFS=\IBMNVDM2\BIN\ANXIFCOM.IFS
```

The `IBMNVDM2.INI` file stores information about the server and the client. If you want to be prompted to enter the client machine name, put a "?" as the client name in the `ClientName` section of `IBMNVDM2.INI`.

8.4.2.2 Contents of IBMNVDM2.INI

The following example shows the contents of the IBMNVDM2.INI file.

```
//*****  
//*  
//* IBM NetView DM/2 Version 2 INI File *  
//* *  
//*****  
ServerName      = SDSERV1  
ClientName      = ?  
AdapterNum      = 0  
AttachTimeOut   = -1  
ReceiveTimeOut  = 15000  
RequestTimeOut  = -1
```

8.4.2.3 Contents of STARTUP.CMD

The following example shows the contents of the STARTUP.CMD file.

```
set endlbpath=%vdisk%\IBMNVD2\DLL;Z:\DLL;  
set path=%path%%vdisk%\IBMNVD2\BIN;Z:\BIN;  
set dpath=%dpath%%vdisk%\IBMNVD2;Z:\;  
%vdisk%  
md \ibmnvdm2  
copy a:\ibmnvdm\*. * %vdisk%\ibmnvdm2\*. *  
md \ibmnvdm2\bin  
copy a:\ibmnvdm\bin\*. * %vdisk%\ibmnvdm2\bin\*. *  
md \ibmnvdm2\dll  
copy a:\ibmnvdm\dll\*. * %vdisk%\ibmnvdm2\dll\*. *  
\IBMNVD2\BIN\ANXPULAG.EXE
```

8.4.3 TME10 SD 3.1.3

For detailed information about TME 10 boot diskettes, please refer also to "Prepare Pristine Client Boot Diskettes (OS/2 Warp 4)" on page 425.

The following illustrates a customized version of the TME 10 SD 3.1.3 boot diskettes. It uses NetBIOS with the ANXIFS as the redirector and starts the installation using the information provided by the NVDM.CFG file.

We assume that the partitioning and formatting of the hard disk is done. If you want to type in the name of the client and of the server at the command line, you should use the statement STARTNB.CMD instead of FNDCMPS.EXE. Have a look at the second version of the STARTUP.CMD.

8.4.3.1 Additional files on Diskette 2

The following files must reside additionally on Diskette 2:

NVDM.CFG SETUP.COMD STARTNB.COMD

In the sample NVDM.CFG, we install the Client LAB1 from the server machine SDSERV1. If you are using the STARTNB.COMD, you can specify the client name, server name and the target drive of the installation. The NVDM.CFG file is then created by the STARTNB.COMD command.

8.4.3.2 Contents of NVDM.CFG

The following example displays the contents of the NVDM.CFG file.

```
WORKSTATION NAME:        LAB1
SERVER:                 SDSERV1 NBI SDSERV1
PROTOCOL:               NBI LAB1 0 2
MESSAGE LOG LEVEL:      D
LOG FILE SIZE:          524288
API TRACE FILE SIZE:    524288
TRACE FILE SIZE:        3000000
DACA RETRY TIME:        30
REPOSITORY:             c:\SOFTDIST\repos
WORK AREA:              c:\SOFTDIST\work
BACKUP AREA:            c:\SOFTDIST\backup
SERVICE AREA:          c:\SOFTDIST\service
CONFIGURATION:          CLIENT
TARGET MODE:            PUSH
MACHINE TYPE:           OS/2
TARGET ADDRESS:         SDSERV1
```

We assume that, as illustrated in "ANXIFS" on page 301, X: is the drive of the product images and Z: for the remote agent.

8.4.3.3 Contents of STARTUP.COMD

The following example illustrates the contents of the STARTUP.COMD file.

```
set target=C:\softdist
set path=%path%z:\
set dpath=%path%z:\
set endlbpath=z:\

md %target%
copy a:\nvdms.cfg %target%
md %target%\repos
md %target%\work
md %target%\backup
md %target%\service
```

```
md %target%\uicfg
md %target%\queue
```

```
z:
fndcmps.exe
```

8.4.3.4 STARTUP.CMD versus STARTNB.CMD

The following example shows that STARTNB.CMD will be executed out of STARTUP.CMD.

```
set path=%path%z:\
set dpath=%path%z:\
set endlibpath=z:\
```

```
startnb.cmd
```

If TPC/IP is to be used instead of NetBIOS, you need to use NFS to access redirected drives. Drives Z: and X: need to be mounted out of the STARTUP.CMD:

```
set target=C:\softdist
set path=%path%z:\
set dpath=%path%z:\
set endlibpath=z:\
...
md %target%
copy a:\nvdcm.cfg %target%
md %target%\repos
md %target%\work
md %target%\backup
md %target%\service
md %target%\uicfg
md %target%\queue
...
mount -u0 -g0 z: 9.3.1.94 d:\cid\os2opsys\softdist
mount -u0 -g0 x: 9.3.1.94 d:\cid\
z:
fndcmps.exe
```

Chapter 9. LAN CID Utility (LCU) / SrvIFS

With the IBM Multi-Protocol Transport Services product, a utility called LAN CID Utility is shipped and abbreviated as LCU. This utility can be easily used to automate installation processes. Not only the installation of OS/2 base operation system can be performed through this utility, but also other CID-enabled products can be installed.

The installation process allows you to set up a procedure to install all the needed products one after another, doing the required reboots as needed. LCU tracks the current state of the installation process across IPLs and ensures that each step is executed in correct sequence. The user or administrator is no longer involved with installation process itself after it has been started for a dedicated machine. The only action required is to launch the installation process by starting the machine with three boot diskettes or enable a connection to the code server machine through the LAN.

If you have a product that is not CID-enabled, you have to evaluate if it is possible to customize it, for example using REXX, and integrate it into LCU installation process.

9.1 Hardware and Software Recommendations

For a dedicated code server, we recommend using a machine that has at least the following features

- A fast bus system like:
 - Micro Channel Architecture (MCA bus)
 - PCI bus, which has become a kind of standard today.
- At least an 80486DX processor
- At least 16 MB RAM, 64 MB is a good size to use
- Fast LAN adapter, Busmaster Card
 - Auto Token-Ring LAN Streamer
 - 100/10 Ethernet Adapter
- A CD-ROM drive
- Fast hard drive, best choice is a RAID 5 for security of data

If you are not using a RAID system, use two physical hard disks: one for OS/2 and the code server's base code, and second disk for the CID and Image directories.

If you use a machine that has other applications running at the same time, you should consider increasing memory.

In an environment with many clients, you may want to use a third physical disk for log files and other files that are written from the clients. Then this disk is the only one to which the clients need write access. It would also be the only disk where you will not know in advance how much space is needed.

The clients' control files and response files require disk space. For one client, these files do not amount to much, but if you intend to install thousands of clients, you must take this into account.

9.1.1 Estimated Space Requirements of Product Images

Table 18. Disk Space Recommendations for Product Images

Product Name	Estimated Space for Images
OS/2 Warp 4	190 MB
eNetwork Personal Communications for OS/2 Warp	11 MB
eNetwork Communications Server for OS/2 Warp	22 MB
DB2 V 2.1.2 including server, Client Application Enabler (CAE), and single user package.	61 MB
LAN Distance	3 MB
LAN Requester	14 MB
MFS	3 MB
MPTS	7 MB
NetFinity Agent	4 MB
NetWare Requester	5 MB
NSC	1 MB
TCP/IP	17 MB
Transaction Server (CICS) for OS/2 Warp	21 MB
CICS clients for OS/2, DOS and Windows	8 MB
MQ Series for OS/2 V2.0.1	30 MB
SystemView Agent	4 MB

9.2 Setting Up the LCU Code Server

This section describes the setup of a code server running LCU as the software distribution manager.

9.2.1 Base Installation

The base operating system is to be installed first from the OS/2 Warp 4 CD-ROM. The standard options are good, but it is not necessary to load the following features in order to gain place:

- OS/2 Tutorial
- Warp Guide Tutorial Agent
- Games
- Multimedia

It is recommended that the disk used for the CID directories be formatted with HPFS. Ensure that disk caching is enabled, for example:

```
IFS=C:\OS2\HPFS.IFS /CACHE:2048 /CRECL:4 /AUTOCHECK:CD
```

Using HPFS386 through OS/2 Warp Server with LCU can improve disk access drastically and therefore increase the speed of multiple remote installations when enough memory is installed at the server and the [filesystem] segment in the \IBM386FS\HPFS386.INI file defines an appropriate amount of memory used for caching.

MPTS must be installed to support the LAN adapter installed on the code server. To run LCU, install NetBIOS either through NetBEUI or over TCP/IP.

9.2.2 Recommended CID Directory Structure

LCU does not require any fixed directory structure for installation, but it is recommended to use a common CID directory structure to have a clear design of the needed files.

It is recommended to use a separate partition or fixed disk for the CID structure with enough space available to hold all the product images, additional fixes, response files, LCU command files, and log files. Be aware that, after every client installation, there will be new files in your directory structure.

The structure illustrated in Figure 3 on page 18 is a good example because it uses almost the same structure that is used on the OS/2 Warp 4 CD-ROM.

9.2.2.1 Building the CID Structure using the WARP40.ZIP Tools

On the OS/2 Warp 4 CD-ROM, you can find a zipped file called WARP40.ZIP stored in the \CID\IMG\MPTS\UTILITY\LCU directory. When unzipping the file, a directory structure is created and sample .LOG, .CMD, .RSP and other configuration files are stored into the according subdirectories.

To unzip WARP40.ZIP file, execute the following command from an OS/2 command line:

```
PKUNZIP2 -d E:\CID\IMG\MPTS\UTILITY\LCU\WARP40.ZIP D:\
```

where **E:** is the drive letter of the CD-ROM drive and **D:** the hard drive where the files to be copied. The **-d** parameter specifies subdirectories to be created.

If you are interested in sample response files, sample log files, a sample SRVIFS configuration file and an LCU command file, you may execute the command mentioned above. However, the recommended directory structure is slightly different from the one created through the WARP40.ZIP unzip command. The next section illustrates the manual setup of the code server.

9.2.2.2 Copying OS/2 Warp 4 CD-ROM to the Code Server

Images of all the products contained in OS/2 Warp 4 and all LCU utilities must be loaded from the CD-ROM into the defined directory structure.

Assuming that the CD-ROM is accessed as **E:** and the CID structure is on **D:**, enter the following commands:

```
XCOPY E:\CID\IMG D:\CID\IMG /S /E
XCOPY E:\OS2IMAGE D:\CID\IMG\OS2WARP4 /S /E
XCOPY E:\CID\SRVIFS D:\CID\IMG\SRVIFS /S /E
XCOPY E:\CID\LOCINSTU D:\CID\LOCINSTU /S /E
XCOPY E:\CID\EXE D:\CID\EXE /S /E
XCOPY E:\IBMINST\NPNWINST.EXE D:\CID\EXE
```

We recommend renaming the \CID\IMG\TCPAPPS directory into \CID\IMG\TCPIP to assure consistency among products.

9.2.3 Copying REXX to the Code Server

GETREXX helps you copying the REXX support necessary for the clients to be able to run the LCU command files:

```
E:\CID\LOCINSTU\GETREXX D:\CID\IMG\OS2WARP4 D:\CID\EXE
```

assuming the **E:** drive is the CD-ROM drive with the inserted OS/2 Warp 4 CD-ROM and **D:** is the drive where REXX-related files are to be copied.

9.2.4 Server-Enabling of the Code Server using SrvIFS

THINSRV will extract the necessary code server files, verify supplied parameters, copy the necessary files to the target and update the CONFIG.SYS and STARTUP.CMD of the code server workstation to automatically start the code server at system start-up.

The following files are installed on the target by THINSRV.

SERVICE.EXE
IFSDEL.EXE
XI1.MSG
XI1H.MSG

The SrvIFS server will share files. They have to be defined in SERVICE.INI file. Before running THINSRV, SERVICE.INI has to be created or updated. Also, SrvIFS requires LAN transport to be installed.

The SERVICE.INI file may contain the lines as shown in Figure 72.

```
;
; SERVICE.INI file for CODESERV
;
NAME=CODESERV
GROUPNAME=NO
ADAPTER=0
MAXCLIENTS=10
MAXFILES=500
CLIENTWORKERS=12
PATH=D:\CID
PERMITWRITE=NO
PERCLIENT=NO
;
; Aliases
;
ALIAS=READONLY,SINGLE,IMG,D:\CID\
ALIAS=READWRITE,SINGLE,LOG,D:\CID\LOG
```

Figure 72. Example of a SERVICE.INI File

Beyond these changes, the server name can be changed from CODESERV to the name you chose.

THINSRV will update the `PATH` and `DPATH` statements of the target's `CONFIG.SYS` file. THINSRV will also add a `START` statement to the target's `STARTUP.CMD` file. Enter the following single-line command:

```
D:\CID\IMG\SRVIFS\THINSRV /R:C:\SRVIFS\SERVICE.INI /T:C:\SRVIFS
/S:C:\CID\IMG\SRVIFS /TU:C:
```

The server can be started immediately either by running the `STARTUP.CMD` file or by running `SERVICE` from the `\SRVIFS` directory.

The following figure displays a `SrvIFS` server initialization file that supports restricted server access:

```
;
; SERVICE.INI file for CODESERV
;
Adapter      = 0
MaxClients   = 5
MaxFiles     = 102
Name         = CODESERV
Groupname    = No
ClientWorkers = 24
Authlist    = CODESERV.LIS
Path        = D:\CID
Perclient    = No
PermitWrite  = No
;
Alias = readonly, single, IMG, D:\CID
Alias = readonly, single, LOG, D:\CID\LOG
;
```

Figure 73. Example of a `SERVICE.INI` File with Limited Client Access Security

The authorization list file, `CODESERV.LIS`, contains NetBIOS names of workstations that are allowed to request remote installations as well as MAC addresses. See Appendix A.1.2, "Example Authorization List File for `SERVICE.EXE`" on page 446 for more information.

9.2.5 Other Information on the Code Server

In some cases, it may be necessary to add some other components to the code server, such as `FixPaks`, `Corrective Service Diskettes (CSD)` or `ServicePaks`.

`ServicePaks`, `FixPaks` and `CSDs` of `OS/2 Warp 4` and `OS/2-related` products should be copied to the `\CID\CSD` directory of the directory tree in its

corresponding subdirectory. For example, the service level of OS/2 Warp 4 FixPak 4 is called XR_M004. FixPak 5 represents service level XR_M005. Therefore, the directories to be created are \CID\CSD\XR_M004 and \CID\CSD\XR_M005, as shown in Figure 74 on page 317.

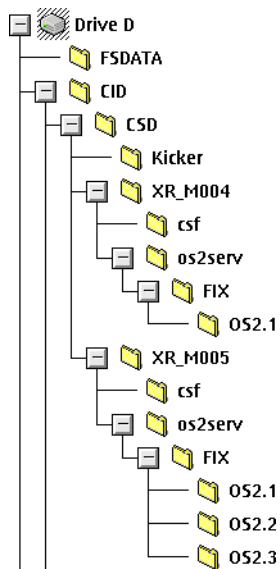


Figure 74. Corrective Service Diskettes Directory Tree

OS/2 base operating system and OS/2-related products FixPaks can be obtained through the Internet at the following Web site:

<http://service5.boulder.ibm.com/pspfixpk.nsf/>

The external Web site for Remote Software Update (RSU) is as follows:

<http://ps.boulder.ibm.com/pbin-usa-ps/getobj.pl?pdocs-usa/softupd.html>

Also check Chapter 3.7, "FixPak Installation" on page 61 for more information about FixPaks and ServicePaks.

Other utilities may be useful to install as those on the CD-ROM enclosed with this manual. Copy the following utilities to the \CID\EXE directory:

- ADDRESS.CMD

This utility is used during NetWare requester installation to fetch the address of the LAN adapter and add it to the MPTS response file. If an address is found in the response file, it will not be changed.

- CHKLANLK.CMD

This utility stores the list of the locked files during installation in order to postpone their processing. It also updates an existing list with a new list of locked files.

- RSTLANLK.CMD

This utility restores the list of a previously stored list of locked file and updates CONFIG.SYS to allow its processing at next boot.

- DISKPRP.CMD

This procedure is an example of what can be done to partition the disk if default partitioning is not sufficient.

- BOOT4LCU.CMD

The use of this procedure is described in Part 9.3.2.1, "The BOOT4LCU Command" on page 320.

9.3 Preparations at the Client Workstations

The following section illustrates client-specific preparations to enable software distribution.

9.3.1 Requirements on the Clients

Please read the the instructions that come with the products as far as remote installation through CID is concerned. In most cases, the remote installation program is different than the installation program for a local install. The client can be on a different LAN segment than the server; however, make sure the clients are technically capable of finding the code server:

- When using NetBIOS, there is a limit of seven bridges or bridging routers to ensure connection.
- When using NetBIOS over TCP/IP, we recommend using a NetBIOS name server, such as NTS Shadow, to ease NetBIOS name to TCP/IP address resolution. There are almost no limits to the number of routers to be crossed to find each other (client and server).

The time it takes to CID install the products depends on the products that you are installing. The faster the processor, disk, file system, and RAM the faster the CID installation process. File services on the server are very CPU-intensive services.

9.3.2 Creating LCU Boot Diskettes

In total, there are four steps to create the set of three OS/2 Warp 4 LCU boot diskettes. These steps can be made done manually as explained in the following topics. A BOOT4LCU.COMD procedure on the CD-ROM distributed with the book integrates the four steps.

1. The first step is to create the set of three plain OS/2 boot diskettes. Use the `SEDISK` command (see Chapter 4.4, "The SEDISK Command" on page 79) to start the creation:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\WARP /T:A:
```

assuming D: is the drive where the product images of OS/2 Warp 4 reside.

PCMCIA Support

When making boot diskettes for ThinkPads or other computers with PCMCIA adapters, pass the `/P` parameter with the `SEINST` command. `/P` describes the type of computer the boot diskettes will be created for and installs the appropriate PCMCIA drivers and socket services.

2. Adding LAN transport to the client diskettes using `THINLAPS`.

In order to transfer NetBIOS and the network drivers onto the diskettes, the code server administrator uses a utility called `THINLAPS`. For a detailed description, refer to Chapter 4.11.1, "The THINLAPS Command" on page 112.

`THINLAPS` will install a seed LAN transport system on the diskettes and update the `CONFIG.SYS` accordingly.

```
D:\CID\IMG\MPTS\THINLAPS D:\CID\IMG\MPTS A:\ [adapter.nif]
```

where D: is the drive where CID image of MPTS resides and `[adapter.nif]` the name of the file which describes the LAN adapter and is located in the `D:\CID\IMG\MPTS\IBMMACCOMMS` directory. For example, the `IBMTOK.NIF` driver is used for most IBM Token-Ring Adapters and IBM-compatible adapters.

3. Install LCU redirector using `THINIFS`.

In order to transfer the SrvIFS redirector code to the diskettes, the code server administrator uses a utility called `THINIFS`. For a detailed description of `THINIFS`, refer to Chapter 4.9.2, "The THINIFS Command" on page 101. `THINIFS` will install the SrvIFS redirector files to the diskettes and update the `CONFIG.SYS` accordingly.

```
D:\CID\IMG\SRVIFS\THINIFS /S:D:\CID\IMG\SRVIFS /T:A:
```

```

/SRV:\\CODESERV\IMG /REQ:CLIENT1 /D:X
D:\CID\IMG\SRVIFS\THINIFS /S:D:\CID\IMG\SRVIFS /T:A:
/SRV:\\CODESERV\LOG /REQ:CLIENT1 /D:Z:

```

where **D:** is the drive where **SrvIFS** resides. **CODESERV** is the name of the code server as defined in the **SERVICE.INI** file, **IMG** the alias where the product images reside and **LOG** the alias where the log files reside as shown previously the **SERVICE.INI** file.

CLIENT1 is the client name given that will log on to the **SrvIFS** code server. The **/REQ:** parameter can have the following values:

- The name of the client which will be the requester's NetBIOS name.
- The requester's name randomly chosen by **SrvIFS** with the ***** value that is passed to the **/REQ:** parameter.
- The user will be prompted to enter a client name at startup time when ***P** is passed to the **/REQ:** parameter.

4. Install LCU client (**CASINSTL**)

CASINSTL installs the LCU agent, creates **STARTUP.COM** and updates the **CONFIG.SYS** file on the diskette.

For a detailed description of **CASINSTL**, refer to Chapter 4.10.1, "The **CASINSTL** Command" on page 107.

```

D:\CASINSTL /TU:A: /CMD:X:\CLIENT /PL:X:\CID\DLL\WARP;X:\CID\LOCINSTU
/PA:X:\CID\LOCINSTU /D

```

9.3.2.1 The **BOOT4LCU** Command

On the CD-ROM enclosed with this redbook, there is a procedure called **BOOT4LCU.COM** which creates the diskettes with a single command. This procedure has to be run on the code server because it will fetch the software to load from the server.

The syntax of this single-line command is:

BOOT4LCU Command Syntax

```

BOOT4LCU /CID:<CIDDir>
          /SRV:<CID Server Name>
          /NIF:<Adapter>
          /P:<PCMCIA Number>

```

where:

<CIDDir>	is the drive where the CID structure resides (default: C:)
<CID Server Name>	is the name of the code server as defined in the SERVICE.INI file (default: CIDSERVER).
<Adapter>	is the name of the adapter.NIF file as defined in the MACS directory of the CID images of MPTS. There is no default for this parameter.
<PCMCIA Number>	is the line number of the PCMCIA laptop defined in PCMCIA.TBL. This table can be found in \OS2\INSTALL directory of an installed OS/2 Warp 4 system. The default is that the workstation to be installed is not a PCMCIA workstation.

If the /NIF parameter is not provided, you will be prompted for the LAN adapter driver to be used.

The procedure will find in the \CID directory the files to use. If there can be an ambiguity on the directories to use, you will be prompted for the directory to use.

You will have to have three formatted diskettes ready to successfully use this procedure. During execution, you will be prompted to change diskettes. The messages are those of the called procedures (SEDISK, THINIFS, THINLAPS, CASINSTL).

SEDISK will ask for three diskettes that have to be labelled:

- OS/2 Warp 4 Installation Diskette
- OS/2 Warp 4 Diskette 1
- OS/2 Warp 4 Diskette 2

A informative message will appear several time to inform that CONFIG.SYS has not been found on the diskette. This is expected. The message prompts you to insert the diskette with the CONFIG.SYS file. Insert the Diskette 1. There will be no check that you have inserted the right diskette. When asked to insert the Diskette 2, insert the diskette labelled Diskette 2. There will be a check for the right diskette. You will have to change the diskettes quite a lot of times because each of the called procedures has to write on both diskettes (THINIFS is called twice).

Note: This procedure will run only if the CID structure is that described in Chapter 9.2, "Setting Up the LCU Code Server" on page 313.

9.4 LCU Command File

The LCU command file or Master Installation Program is a means by which an administrator can chain together a number of CID-enabled products as a single installation process on a client workstation. The LCU command file is REXX-based.

DEFAULT.CMD loaded while unzipping WARP40.ZIP is not going to be used because it has been made for a remote installation using the CD-ROM. Instead, we will use the DEFAULT.CMD file loaded from the CD-ROM distributed with this manual.

9.4.1 General information on Command Files

The command files are prepared for a large number of installations. Each installable product gets an index number. These index numbers are generated by the counter variable 'i'. You will have to change the index numbers and adjust the number of install programs accordingly if you do not use the dynamic indexing with the i-variable.

The LCU process tracks the current state of the installation across IPLs and ensures that each CID installation program executes in the correct sequence. Return codes passed from the various installation programs are evaluated for problems, and passed to the administrator when intervention is required. The LCU process also provides a means by which product-specific response files are selected. Once the installation sequence has been put into the LCU command file, the installation process will run at the client workstation with a minimum of human intervention. When the LCU process is started from a client workstation with the boot diskettes, then a lightly attended installation will take place, and an unattended installation occurs when started from a client's hard disk with an OS/2 operating system already installed.

9.4.1.1 LCU Return Codes Processing

The LCU command file is a REXX ".CMD" program that processes good and bad return codes for the CID-enabled install program and reboots of the system and environment variables. Conditional logic is imbedded within the LCU command file to handle different return codes and environment variables.

A CID-enabled install program returns four types of return codes on which LCU can act.

The four return codes are:

- Successful completion, reboot not required

- Successful completion, reboot
- Successful completion, reboot and call me back
- Error

LCU manages the state of the INSTALL product by validating its state as returned by the product install program. Note the following steps:

- When a product install program requests a reboot with callback, it is the responsibility of the exiting product install program to set the right byte (xx may vary from 00 to FF) of the return code to its next install state.

For example, on the initial call from LCU, the state variable is x'00', and it may be incremented (by one) by the product install program for each reboot request that will return to the currently exiting product install program.

- LCU validates that the Product Install state is different than the last time the product install program was invoked.
- LCU reboots the workstation, retrieves the product's install state parameter, remembers it and passes it to the invoked product install program through the `REMOTE_INSTALL_STATE` state variable.

LCU Reboot

CID-enabled install programs have the ability, through return codes, to request that LCU queues a reboot of the workstation. In LCU, if a reboot was queued by a program, the reboot does not necessarily happen immediately, but will happen after the next "Call CheckBoot" is encountered.

If a "Call CheckBoot" is encountered and a reboot was not queued by any of the programs since the last reboot, the workstation will not reboot and will continue to the next state.

The following steps describe the LCU REBOOT processing:

- Product install programs communicate to LCU that a reboot is required by specifying return code x'FE 00' upon exit to LCU.
- The product install return code is used by LCU to set a state variable `REMOTE_INSTALL_STATE`, which is in ASCII format (1 to 3 characters depending on the size of the value, from 1 to 255). The `REMOTE_INSTALL_STATE` variable is saved by LCU in `CONFIG.SYS` before LCU causes a physical reboot of the workstation to occur. The variable is saved as an OS/2 environment variable so that after the reboot, LCU can interrogate it again.

- LCU agent code gains control on reboot because of the command line placed into the STARTUP.CMD file of the workstation boot drive and executes the LCU REXX program residing on the code server disk.
- The saved state variable is interrogated by LCU to detect infinite loops and for product install programs to determine their execution state.

LCU Reboot and Callback

CID-enabled install programs have the ability, through return codes, to request that LCU call them back after a reboot of the client workstation. This is a combined return code "queue a reboot and call me back". Just as in the case of queue reboot, the reboot will not happen until the next "Call CheckBoot" is encountered. If an install program requests to be called back, LCU will not progress to the next state after the reboot; the request will be honored, LCU will enter the same state it was in before the reboot, and it will re-execute the install program that requested to be called back. All install programs in the same state, and which have state variables that did not request to be called back, will not be executed again. All install programs in the same state, and which do not have state variables, will be executed again. So be aware of this behavior when you install not only that product that requires to be called back in this section but also some other products without state-variables. It may cause you some problems; so it can be easier to install a program that requires a reboot in a separate section.

9.4.2 Working with Default Response Files and LCU Command Files

LCU can do automatic selection of LCU command files and response files based on the client name that is calling the code server.

Default LCU Command File

LCU does an automatic command file selection based on the LCU command file name. The selection is done in two steps:

1. `CASAGENT` looks for its command file in the `CLIENT` directory where all client command files are located. `CASAGENT` will check the directory for an LCU command file named `<client name>.CMD`. If it exists, this `<client name>.CMD` will execute.

If it does not exist:

2. If the `/D` parameter is used, `CASAGENT` will search for a LCU-command file named `DEFAULT.CMD` in the directory specified by the `/CMD:` parameter. If the `/D:` parameter is used, `CASAGENT` will search for the LCU command file named together with this parameter

If none of the files exist, `CASAGENT` will exit and end the installation.

The code server administrator has the following choices:

- Build a unique LCU command file for each client workstation
- Build a default LCU command file for all client workstations
- Build a unique LCU command file for selected client workstations and a default for all other client workstations

It is recommended to build a default LCU command file for all client workstations and build only unique LCU command files for selected workstations. By doing this, the code server administrator can create common boot diskettes for all client workstations where the user is asked to type in the client workstation name. If a particular client workstation needs a specific LCU command file, the administrator can create a new LCU command file and give it a particular client name. The administrator tells the user the new name to use and if the user correctly enters that name the <client name>.CMD will execute. The administrator can also decide to give the user an LTS diskette with a correctly coded client name. If there is no corresponding <client name>.CMD stored on the code server, the DEFAULT.CMD will be executed anyway.

Default Response File

LCU can also do automatic default response file selection. The code server administrator can decide if a CID product install program will use a specific response file based on the client name or use a default response file.

It is also recommended to build a default response file for all client workstations and build only unique response files for selected workstations. This is recommended but not always so easy because of the hardware differences between the different client machines. The way to resolve this is to generate default response files with the common keywords for all clients. The individual settings are defined in response files for the different clients or group of clients that can be merged into the default response file using the `INCLUDE` keyword statement. The merging process can be done as a default step before any installation starts. This process scans through the response files and replaces all variables in the response file that point to the client name. By using `INCLUDE` and variable techniques, you can reach the highest level of automation in the CID environment.

Another way to implement the differences between the workstations is to create a new response file and give it a particular client name. The administrator tells the user the new name to use, and if the user correctly enters that name, the <client name>.RSP will be selected by the CID install program. The administrator can also decide to give the user a set of boot diskettes with a correctly coded client name.

9.4.3 LCU Command File Structure

The LCU REXX command file is composed of three basic sections.

First Section: Variables Definitions

The first section contains variables. For each of the products you want to install with LCU, you must configure here each of the install programs. This section contains the path to the install programs, the parameters to be used by the install programs, the path to the response file, and the default response file. You may **NOT** modify any line after the remark `DO NOT MODIFY THE NEXT EIGHT LINES. Modifications MUST only start after the remark MODIFICATIONS START HERE.`

Global Variables: Global variables allow the identification of an object to the command file once and refer to it later with the variable name. The following two statements need to be modified with the system information.

```
bootdrive='C:'
configsys = bootdrive ' \CONFIG.SYS'
exepath = 'X:\EXE\OS2'
```

Please take care to ensure that the `exepath` really points to the OS/2 version that will be installed on the client if your CID structure include other versions of OS/2 than OS/2 Warp 4.

Product Data Section: The following statements are product-specific data. Each product installed needs a set of these statements. The program-specific parameters are linked together through the "comma" at the end of each statement. This example is for an OS/2 operating system install using `SEINST`.

```
x.seinst = 7
x.7.name='WARP '
x.7.statevar = 'CAS_' || x.7.name
x.7.instprog = 'x:\exe\os2\seinst ',
  '/b:' || bootdrive || ' ',
  '/s:'cidingdir'\warp ',
  '/t:c:\service ',
  '/ll:Z:\warp\ ' || client || '.log ',
  '/r:'
x.7.rspdir = 'x:\rsp\warp'
x.7.default = 'default.rsp'
```

Each product is defined with its installation program and parameters in a variable as described above. To make it easy to delete or add a product from or to this section, we did not use absolute numbers in the variable name. We

used a counter variable 'i' that increases for every product. The variable NUM_INSTALL_PROGS is set equal to this counter.

```
i=i+1
x.mpts = i
x.i.name='MPTS Installation'
x.i.statevar = 'CAS_' || x.i.name
x.i.instprog = cidimgdir'\mpts\mpts ',
' /e:maint ',
' /s:x:\img\mpts ',
' /t:' || bootdrive || '\ ',
' /l1:Z:\mpts\' || client || '.log ',
' /r:'
x.i.rspdir = 'x:\rsp\mpts'
x.i.default = 'default.rsp'
```

The Table 19 on page 328 describes the product variables.

Table 19. Product Variable Description

Variable	Title	Description
x.seinst	Structure index	This contains the name of the install program and a number to identify the program.
x.i.name	Product name	A user-defined name for the product. This name must be unique for each of the install programs. It is used for messages and for building the value of x.i.statevar.
x.i.statevar	State variable name	The name of the environment variable that will be used to maintain the install state of the product across reboots. This variable is constructed from the product name. Note: The <code>statevar</code> keyword MUST always be defined (not specified is indicated by a NULL string).
x.i.instprog	Fully qualified install program name and parameters	The name of the install program for this product with its path and all the specific parameters for the install.
x.i.rspdir	Response file directory	The path to the response files for this product.
x.i.default	Default response file name	The name of the default response file to be used if the one for this client cannot be found.

Default Response File: The LCU command file can do automatic default response file selection. The program will check the directory specified in `x.i.rspdir` for the `<client name>.RSP`. If it exists, the fully qualified path to this response file will be appended to the `instprog` string. If it does not find it, the fully qualified path to the default response file specified in `x.i.default` will be appended to the `instprog` string. The program does NOT check that the default response file exists.

If you want LCU to do default response file selection automatically for you, you must put the `/r:` parameter at the end of the parameter list without any trailing blanks. Then specify the response file directory in `rspdir` and the default response file in `default`.


```

x.seinst = 7
x.7.name='WARP '
x.7.statevar = 'CAS_' || x.7.name
x.7.instprog = 'x:\img\os2warp4\seinst ',
' /b:' || bootdrive || ' ',
' /s:'cidingdir'\os2warp4 ',
' /t:c:\service ',
' /l1:Z:\os2warp4\ ' || client || '.log ',
' /r:'
x.7.rspdir = 'x:\rsp\os2warp4'
x.7.default = 'default.rsp'

```

If you wish to hard code a specific response file, you must set `rspdir` and `default` to `'`. (`'` indicates a NULL string; see example below).

```

x.seinst = 7
x.7.name='WARP '
x.7.statevar = 'CAS_' || x.7.name
x.7.instprog = 'x:\img\os2warp4\seinst ',
' /b:' || bootdrive || ' ',
' /s:'cidingdir'\os2warp4 ',
' /t:c:\service ',
' /l1:Z:\os2warp4\ ' || client || '.log ',
' /r:specific.rsp'
x.7.rspdir = ''
x.7.default = ''

```

Product Count: The last line of the first section indicates the total number of products initialized in the product data section.

```
NUM_INSTALL_PROGS = 21
```

When you add a new program in the product data section, you must set `NUM_INSTALL_PROGS` to the total number of programs initialized. If you use the counter variable technique mentioned above, the product count is done implicitly by increasing the variable.

```
NUM_INSTALL_PROGS = i
```

Additional SRVATTCHs: If you are using SrvIFS for redirection, a modification that can be made is to add a certain number of additional `SRVATTCHs` to the code server or to any other servers. These `SRVATTCH`

statements can be added before or after the global variables. For example, they could look like this:

```
'SRVATTCH S: \\SERVER1\ALIAS'  
'SRVATTCH T: SERVER2'
```

By using additional `SRVATTCH` statements in the LCU command file, the administrator can connect the client workstation to different drive aliases defined on the same code server or on any other SrvIFS server located on the same logical LAN.

Second Section of LCU Command File

The second section of the LCU command file contains the install statements. Depending on the products to be installed, there can be several phases in the total install. Most programs require a reboot after being installed. This section sets up the steps needed and ensures the reboots happen when they are needed.

Here is an example of the second section:

```
Do Forever  
  Select  
    when OVERALL_STATE = 0 then do  
      if RunInstall(x.seinst) == BAD_RC then exit  
      if RunInstall(x.MPTS) == BAD_RC then exit  
      if RunInstall(x.thinifs1) == BAD_RC then exit  
      if RunInstall(x.thinifs2) == BAD_RC then exit  
      if RunInstall(x.casinst1) == BAD_RC then exit  
      Call CheckBoot  
    end  
    when OVERALL_STATE = 1 then do  
      if RunInstall(x.ibmpeer) == BAD_RC then exit  
      Call CheckBoot  
    end  
    when OVERALL_STATE = 2 then do  
      if RunInstall(x.ifsdel) == BAD_RC then exit  
      if RunInstall(x.casdelet) == BAD_RC then exit  
      Call Reboot  
    end  
  end  
end  
exit
```

In this example, at the beginning `OVERALL_STATE` variable is set to 0, so `RunInstall` function is run to install the product defined in section 1 by variable `x.seinst` (for instance: OS/2 WARP).

RunInstall builds the installation command of the product to be installed, executes it, analyzes the return code to set up the environment variables, and returns a GOOD_RC or BAD_RC code. Environment variables can be set up to ask for a reboot or not and for a call back of the installation program or not.

Here are some more details on the execution of this previous example:

1. When OVERALL_STATE = 0 then do

This statement indicates to the LCU command file to execute the statements between this statement and the corresponding "end" statement whenever the OVERALL_STATE is equal to 0.

2. If RunInstall(x.seinst) == BAD_RC then exit

The first time through this state, this statement will install the base operating system. SEINST is checking the boot drive. If the installation was started with boot diskettes or RIPLed, SEINST will ignore the /T: parameter; even if /T: is C:\SERVICE, it will be ignored.

SEINST Return Codes

SEINST will issue return code x'FF02' upon exit if booted from diskette.
SEINST will issue return code x'FF01' upon exit if booted from fixed disk.

SEINST will request a reboot and to be called back. The first time through this state SEINST will request a call back by using return code x'FF02' because the installation was booted from diskette.

3. If RunInstall(x.mpts) == BAD_RC then exit

This statement will install MPTS for the production system. The boot drive's CONFIG.SYS is modified in this step. This program will request a reboot and will not request to be called back.

4. If RunInstall(x.thinifs1) == BAD_RC then exit

This statement will install the LCU redirector. LAN connectivity to the code server is added to the boot drive CONFIG.SYS in this step. THINIFS1 will attach to the code server alias. THINIFS will update the boot drive CONFIG.SYS located by the value defined for the /TU: parameter. The following statements are added to the CONFIG.SYS:

```
DEVICE=target\SRVIFS.SYS
IFS=target\SRVIFSC.IFS <options>
CALL=target\SRVATTCH.EXE drive_letter: shared-directory
```

In addition, the PATH statement is also updated to include the target of the installation. This program will request a reboot. Please refer to Chapter

4.9.2, "The THINIFS Command" on page 101, for a description of the THINIFS parameters.

5. If RunInstall(x.thinifs2) == BAD_RC then exit

This statement will install the LCU redirector again. THINIFS executes twice in order to attach to a LOG redirected drive prior to the invocation of the LCU command file. LAN connectivity to the code server is added to the boot drive CONFIG.SYS in this step. THINIFS2 will attach to the code server LOG alias. THINIFS will update the boot drive CONFIG.SYS located by the value defined for the /TU: parameter. The following statement is added to the CONFIG.SYS:

```
CALL=target\SRVATTCH.EXE drive_letter: shared-directoty
```

The following two statements are not added because they have been added by previous call to THINIFS:

```
DEVICE=target\SRVIFS.SYS  
IFS=target\SRVIFSC.IFS <options>
```

6. If RunInstall(x.casinst1) == BAD_RC then exit

LCU is installed in this step. SRVREXX is added to the bottom of the boot drive CONFIG.SYS along with additional paths added to the DPATH and LIBPATH. CASAGENT is also added to the boot drive STARTUP.CMD.

7. Call CheckBoot

At this point, the LCU command file will check to see if any programs requested a reboot since the last boot. Also, it will check to see if any programs have requested to be called back. The programs SEINST, LAPS, THINIFS1 and THINIFS2 requested a reboot, but SEINST also requested a call back. The OVERALL_STATE variable CAS_STATE is set to 0 so that when the workstation is rebooted, the LCU command file will enter the same state again and re-execute the installation programs. SEINST asked to be called back by issuing return code x'FF02'; therefore LCU is setting its state variable CAS_WARP=2, so that after the reboot, SEINST knows that it is entering this state because it asked to be called back.

8. When OVERALL_STATE = 0 then do

After the reboot this environment variable is tested. All statements between this statement and the corresponding "end" will be executed because the value is equal to 0.

9. If RunInstall(x.seinst) == BAD_RC then exit

The second time through this state, SEINST will do nothing because it knows by looking at the REMOTE_INSTALL_STATE CAS_WARP=2 that the initial installation was booted from diskette. The /T: is not checked and SEINST

will wait until all icons appear on the Workplace Shell. This time, `SEINST` will not request a reboot and will return the successful completion, reboot not required return code `x'0000'` to the LCU command file.

10. `if RunInstall(x.mpts) == BAD_RC then exit`

The second time through this state, this statement will do nothing. This program did not request to be called back the first time, and this program has a state variable indicated in the product data section.

Note

Remember that programs having a state variable defined will never run again the second time the LCU command file encounters them.

11. `If RunInstall(x.thinifs1) == BAD_RC then exit`

`If RunInstall(x.thinifs2) == BAD_RC then exit`

`If RunInstall(x.casinst1) == BAD_RC then exit`

The second time through this state, these programs will be installed again. This is done even though it did not request to be called because it does not have a state variable indicated in the product data section. They will request a reboot.

12. `Call CheckBoot`

At this point, the LCU command file will check if any programs have requested a reboot since the last boot. Also, it will check if any programs have requested to be called back. None of these programs have requested to be called back, but `THINIFS1` and `THINIFS2` have requested a reboot. The `OVERALL_STATE` variable `CAS_STATE` is set to 1 so that when the workstation is rebooted the LCU command file will enter in the next state

`OVERALL_STATE=1.`

13. `When OVERALL_STATE =1 then do`

`If RunInstall(x.ibmpeer) == BAD_RC then exit`

`Call CheckBoot`

These statements indicate to the LCU command file to execute the statements up to the "end" statement whenever the `OVERALL_STATE` is equal to 1.

After installation of `IBMPEER`, `CheckBoot` will test if reboot is necessary. In this case, it is yes. After reboot, the `OVERALL_STATE` variable will be set to 2.

14. `When OVERALL_STATE =2 then do`

This statement indicates to the LCU command file to execute the statements up to the "end" statement whenever the `OVERALL_STATE` is equal to 2.

15.If `RunInstall(x.ifsdel) == BAD_RC` then exit

This statement will remove the LCU redirector statements from `CONFIG.SYS` and erase LCU redirector code from the fixed disk. `IFSDEL` will not remove itself from the system. Path statements from the `PATH`, `DPATH` or `LIBPATH` will not be removed. This program will request a reboot.

16.If `RunInstall(x.casdelete) == BAD_RC` then exit

This statement will remove `SRVREXX.EXE` and the `PATH` and `DPATH` additions that were made before to the `CONFIG.SYS`. It will also remove `CASAGENT.EXE` from `STARTUP.CMD`.

17.Call `Reboot`

This statement will reboot the machine, and this is the last reboot. When the machine reboots, the programs are successfully installed.

Third Section of LCU Command File

The third section contains REXX subroutines for processing the installs. The user WILL NOT make any modifications to this section of the LCU command file.

9.4.4 BASE.CMD Command File

This file was built to be as similar as possible of the local installation process as described in Chapter 3, "OS/2 Warp 4 - Installation Steps Local Install" on page 39.

It is made to the same kind of installation as a local installation does. It allows you to partition your disk as you want, make all the selections that OS/2 response file's keywords allow, install any ServicePak that can be delivered and install selected products of the BonusPak.

Choice of products to install can be made by indicating the name of the products to install in the variable `products_to_install`:

```
/* keywords for products to install : */
/* NW MFS IBMPEER NSC TCPIP LDR NETFIN SVAGENT CLIFI DSPLY */
products_to_install=' IBMPEER TCPIP NW MFS NSC LDR NETFIN SVAGENT CLIFI
DSPLY '
```

MPTS is always installed.

During the first step of the installation, `x.diskprp` is run. The called procedure `DISKPRP` looks for a partition called "WARP". If found, it ends. If not, it creates partitions according to the definitions given within it and reboots the system.

Note: A copy of the `DISKPRP.CMD` file is on the enclosed CD-ROM. `DISKPRP` is illustrated in Chapter A.4, "Partitioning Utility: `DISKPRP`" on page 460.

```
when OVERALL_STATE = 0 then do
  if RunInstall(x.diskprp)      == BAD_RC then exit
  if RunInstall(x.seinst)      == BAD_RC then exit
  if RunInstall(x.fixpak)      == BAD_RC then exit
  if RunInstall(x.mpts)        == BAD_RC then exit
  if RunInstall(x.thinifs1)    == BAD_RC then exit
  if RunInstall(x.thinifs2)    == BAD_RC then exit
  if RunInstall(x.casinst1)    == BAD_RC then exit
  Call CheckBoot
end
```

Then `x.seinst` is run. `SEINST` is the installation program of OS/2 Warp 4. It will install it according to the values found in the response file.

In this example, an OS/2 FixPak is to be installed. Installing it at that time is a good choice because there is no OS/2 locked file. It saves reboot time.

`MPTS` is then installed (`x.mpts`). It is necessary to have the LAN support to link to the code server after the first reboot.

`x.thinifs1`, `x.thinifs2` and `x.casinst1` are the utilities which install the `SrvIFS` and `LCU` codes on the disk for the following steps of installation.

In the second step of installation, the other products included into OS/2 Warp 4 will be installed according to the choice given in the `product_to_install` variable.

```
when OVERALL_STATE = 1 then do

  if pos('DSPLY', products_to_install) >0 then
    if RunInstall(x.dspinst) == BAD_RC then exit

  if pos('CLIFI', products_to_install) >0 then
    if RunInstall(x.clifi) == BAD_RC then exit

  if pos('SVAGENT', products_to_install) >0 then do
    if RunInstall(x.svagent) == BAD_RC then exit
    'call x:\cid\exe\chklanlk.cmd ' bootdrive
  end
```

```

if pos('NW', products_to_install) >0 then do
  if RunInstall(x.nwinst)    == BAD_RC then exit
  if RunInstall(x.nwaddress) == BAD_RC then exit
  if RunInstall(x.nwmpts)    == BAD_RC then exit
end

if pos('MFS', products_to_install) >0 then
  if RunInstall(x.mfs)      == BAD_RC then exit

if pos('IBMPEER', products_to_install) >0 then do
  if RunInstall(x.os2peer)  == BAD_RC then exit
  'call x:\cid\exe\chklanlk.cmd ' bootdrive
end

if pos('NSC', products_to_install) >0 then
  if RunInstall(x.nsc)      == BAD_RC then exit

if pos('TCPIP', products_to_install) >0 then do
  if RunInstall(x.tcpapps)  == BAD_RC then exit
  'call x:\cid\exe\chklanlk.cmd ' bootdrive
end

if pos('LDR', products_to_install) >0 then do
  if RunInstall(x.ldr)      == BAD_RC then exit
  'call x:\cid\exe\chklanlk.cmd ' bootdrive
end

if pos('NETFIN', products_to_install) >0 then do
  if RunInstall(x.netfin)   == BAD_RC then exit
  'call x:\cid\exe\chklanlk.cmd ' bootdrive
end

'call x:\cid\exe\rstlanlk.cmd 'bootdrive

Call CheckBoot                      /* Reboot if it was requested */
end

```

When installing some products, some files cannot be updated because they are already in use. These files are stored in a temporary directory to replace the locked files at the next boot. `CHKLANLK` is a procedure that appends such files to those already stored by a previous product installation and modifies the `CONFIG.SYS` file in such a way that the following products will be able to install. `RSTLANLK` restores the list of these files so they will be installed at the next reboot.

NetWare requester installation is done by three programs. `NPNWINST` is the real installation program. The third program installs the NetWare Requester Support protocol using the `NWMPTS.RSP` response file. This response file is modified by the second procedure, `ADDRESS.CMD`. This procedure fetches the adapter address and adds it to the `NWMPTS.RSP` file if none is found in it.

The third step of installation cleans the disk from the `SRVIFS` and `LCU` components, then reboots the system to finish.

```
when OVERALL_STATE = 2 then do
  if RunInstall(x.ifsdel) == BAD_RC then exit /* Delete SrvIFS req. */
  if RunInstall(x.casdelete) == BAD_RC then exit /* Delete LCU */
  Call Reboot /* Reboot */
end
```


Chapter 10. NetView Distribution Manager/2 (NVDM/2)

This section describes the series of tasks required to enable the automated installation of CID-enabled products in a LAN environment using IBM NetView Distribution Manager/2 Version 2.1. The description is based on NetView DM/2 V2.1, but it is also valid for IBM NetView Distribution Manager/2 Version 2.0. NetView DM/2 V2.1 is required if the server is running OS/2 Warp 3 or higher.

The enhancements of Version 2.1 are mainly in the remote administrator function. The tasks for preparing a server workstation to initiate and control installations remained unchanged, although you may receive slightly different messages. On Diskette 25 of NetView DM/2 V2.1, there are sample change file profiles, response files and procedures. You may want to check on these samples before creating your own files.

Ensure that you have the latest version, CSD and FixPaks installed and the latest NetView DM/2 images loaded.

10.1 NetView DM/2 Overview

NetView DM/2 is designed to centrally control system and application software installation, software change management and data distribution across networks. The objective of this section is to outline some of the features of NVDM/2 in a NetBIOS environment and to give some examples of installing OS/2 Warp V4 and some of its products. It is assumed the reader has some experience with NVDM/2 or has the product manuals available to use.

10.1.1 Components of NetView DM/2

The two major components of NetView DM/2 are the Change Distribution Manager (CDM) and the LAN Download Utility (LDU).

10.1.1.1 Change Distribution Manager (CDM)

This component of NetView DM/2 provides software and data distribution and change-control functions within a domain. These features are included in the Extended Base feature of NetView DM2.

The primary function of CDM is to distribute data and software. You can:

- Start, control and stop the CDM service
- Manage a CC Domain

- Prepare an object for distribution
- Install and maintain software
- Manage remote destinations
- Distribute objects in an APPC network

In our environment, we use the function to manage a CC Domain, prepare an object for distribution, and to install and maintain software. We are not concerned about the functions related to APPC networks.

10.1.1.2 LAN Download Utility (LDU)

The LDU is an alternative to the CDM which provides software and data distribution, but no change control. It requires the installation of the Entry Base of NetView DM/2 on a server machine called the *distributor*.

Any other client in the LAN has the Entry Client feature of NetView DM/2 installed and is a so-called *receiver*.

This implementation is used if you want fast download service for one or more selected files on one or more target workstations in the same LAN. It is possible to download an operating system and applications.

LDU will not be used for the scenarios described in this book. For further information about the LCU component of NetView DM/2, please refer to the NetView DM/2 documentation.

10.1.2 NetView DM/2 Workstation Types

In a LAN environment, the NetViewDM/2 Extended Package base feature must be installed on one of the workstations running OS/2. This machine is the *change control server*. Other workstations in the LAN become *change control clients* with the NetView DM/2 Extended Package client feature installed.

- NetView DM/2 Change Control Server (CC server)

This workstation holds the product images and response files. It also processes the change files that are built specifically for product installations. The CC server manages the installation process, maintains a database of installed products and can provide status reports back to a focal point.

A so-called preparation site workstation, which is often on the same system as the CC server, prepares all images, response files and change files. In a stand-alone LAN environment, NetView DM/MVS is not involved.

- NetView DM/2 Change Control Client (CC client)

These workstations are ready and waiting to execute installation requests which are queued at the NetView DM/2 CC server and initiated by the administrator. The NetView DM/2 CC clients are unattended systems. In a 'pristine' environment, a minimal Netview DM client is installed using the last boot diskette (see "Creation of Boot Diskettes" on page 354).

The group of CC clients controlled by and including the CC server itself is known as the *change control, or CC domain*.

Connections between different LANs are also possible using APPC links between the CC servers and, if required, intermediate nodes and systems running NetView DM/MVS. Within the scope of this book, we only discuss the OS/2 part in a stand-alone LAN domain using NetBIOS.

NetView DM/2 has a utility to be used for preparing the software for distribution. This so-called *builder* creates a change file for each new software package or software update. A change file consists of files that belong to the software product, application, update or program fix, and files that contain information needed for the installation process. Please refer to Chapter 10.9, 'Creating Change Files' on page 361 for more information.

10.2 Prerequisites

Before installing NVDM/2 on the CC server, the following prerequisites are required:

1. OS/2 Version 3 or higher, MPTS and DB2/2 (Version 2.11 is recommended) must be installed and configured properly.
2. CID directory structure with loaded product images accessed locally or remotely, for example, from a file server.

10.2.1 NetBIOS Resource Considerations

The following table refers to the resources required by NetView DM/2 in addition to those required by other applications using NetBIOS.

These resources are defined in the PROTOCOL.INI file. To alter these values, edit this file or use MPTS to change the values.

The Keywords `MaxClients` and `MaxRequests` are defined in the IBMNVDM2.INI file.

<code>Maxclients</code>	Defines the maximum number of clients that can process installation requests concurrently. (Range = 1 to 100).
-------------------------	--

MaxRequests Is the maximum number of threads that NVDM/2 uses to process CC client file access requests. (Range = 1 to 8)

Table 20. NetBIOS Resources for NetView DM/2

	CC Server	CC Client
Sessions	2 + MaxClients	2
Commands (NCBs)	29 + MaxRequests	14
Names	7	6
(GDT) Selectors	at least 10	Default
Datagram packets	15	Default
NetBIOS Timeout	2000	Default
NetBIOS Retries	15	Default

MPTS sets the values for sessions, names and NCBs to the maximum dependent on the adapter. If using APPC sessions to NVDM/MVS or to another CC server, increase the link stations parameter in the 802.2 section by 2+MaxClients.

The values for NetBIOS Timeout and NetBIOS Retries should be set if there are more than 20 clients connected to the CC server. For more information on the resources required, see the *IBM Netview Distribution Manager/2.1 Installation and Customization Guide*, SH19-6915'

10.2.2 Adding Network Adapters to MPTS

If you need to add and configure a LAN adapter that is not included in the standard MPTS installation (for instance IBMTR.OS2 - PCI Token-Ring Adapter), complete the following steps:

1. Make sure PKZIP2 and PKUNZIP2 utilities are available and executable from an OS/2 command line.
2. Create a temporary subdirectory, such as TEMP, and change to the D:\SHAREA\IMG\MPTS\IBMCOMM\MACS directory. Unpack the MACS.ZIP file to the TEMP directory.

```
PKUNZIP2 -d D:\TEMP
```

(Check the parameters for PKUNZIP2/PKZIP2 by entering PKUNZIP2 on the command line from the appropriate subdirectory, for instance C:\CMLIB\PKUNZIP2).

3. Copy your additional NDIS device driver files to this subdirectory (such as IBMTRP.OS2 and IBMTRP.NIF) and erase the file MACS.ZIP.
4. Pack all the files in the TEMP subdirectory to MACS.ZIP

```
PKZIP2 MACS.ZIP *.*
```

5. Erase all the .NIF and .OS2 files so that MACS.ZIP is the only file in this subdirectory. Copy this MACS.ZIP back to the \SHAREA\IMG\MPTS\IBMCOM\MACS subdirectory and delete it from the TEMP directory.
6. If your new NDIS driver provides additional message (.MSG) files, you need to add these files to IBMCOM.ZIP, in the \SHAREA\IMG\MPTS\IBMCOM subdirectory. Unpack IBMCOM.ZIP to the TEMP directory.
7. Copy the additional .MSG files to this directory and delete the IBMCOM.ZIP file.
8. Pack all the files in the TEMP directory back into IBMCOM.ZIP and copy this file back to the \SHAREA\IMG\MPTS\IBMCOM directory.
9. Erase the TEMP directory.

10.2.3 Database 2 for OS/2 Considerations

For better performance of your NetView DM/2 system, you should customize DB2 in case you are using a DB2 for OS/2 version lesser than 2.11. Increase the number of maximum shared segments (`SGLSEG`). The default value is normally set to 25; you should change this value to 40, and this should decrease the time it takes NetView DM/2 to access its tables.

If you do this change after the installation of NetView DM/2, you have to reinstall NetView DM/2 with the **Configuration only** option to load the new value for the use of NVDM/2

To modify the `SGLSEG` value, do the following from an OS/2 command line.

1. Logon locally with SYSADM (System Administrator; a locally defined administrator through User Profile Management) authority.
2. From an OS/2 window, enter the `STARTDBM` command.
3. Next, enter the following command:

```
DBM UPDATE DATABASE MANAGER CONFIGURATION USING SGLSEG 40
```

4. Then enter the `STOPDBM` command to stop the database manager.

After you have made the changes to DB2, restart the database server by executing the `STARTDBM` command..

If you customize the DB2 configuration to restrict access to the database to certain groups or users, please reflect these changes in NetView DM/2, because NetView DM/2 needs a user ID with DBADM authority to its databases.

10.2.4 User Profile Management (UPM)

User Profile Management for OS/2 provides three levels of authority:

- Administrator
- Local Administrator
- User

DB2 Database Manager provides two levels of authorization:

SYSADM The highest authority granted. Allows you to create or erase databases, grant access to databases and change database configuration files.

DBADM Database administrator
Allows access and control of an existing database

Local administrator or administrator access in UPM allows you SYSADM authority to the DB2 database.

SYSADM access to the DB2 databases is required for the installation of NVDM/2. DBADM is needed for the maintenance of the databases, and USER access is required in order to start NVDM/2.

10.2.4.1 Add a new Local Administrator for NetView DM/2

1. Logon locally with an administrator user ID. The default is USERID/PASSWORD. OS/2 Warp Connect or OS/2 Warp 4 user must enter the user ID and password defined at installation time.
2. Select **User Profile Management Services** from the desktop.
3. Select **User Profile Management**.
4. From the menu bar, select **Manage**; then from the pull-down menu, select Manage Users.
5. **Select Add a new User ID** from the Actions menu.
6. For User Type, select **Administrator** (needed for installation of NetView DM/2) or **Local Administrator** (needed for running NetView DM/2) and

complete the required information. For example, the new user ID to be created is NVDM2ADM with the password of PASSWORD.

10.2.5 Modifying STARTUP.COMD Before Installing Netview DM/2

Before you install NetView DM/2, make sure you are logged on locally with the user ID you created in the previous step. To make sure you are always logged on locally with the correct user ID, we recommend creating a STARTUP.COMD with the following statements:

```
@ECHO OFF
LOGON NVDM2ADM /P:PASSWORD /L
STARTDBM
```

where NVDM2ADM is the user ID with the password of PASSWORD. The /L parameter indicates a local logon to the workstation.

After NetView DM/2 was installed, re-edit the STARTUP.COMD file to correct the logon statement since the installation program adds the statement

```
LOGON USERID /P:PASSWORD /L
```

which would not be correct in this scenario. You really do not want to be logged on locally with the default settings because it would cause a security gap in your system.

10.3 Installing and Configuring NetView DM/2

There are two ways of installing NetView DM/2. Either attendedly, by using Presentation Manager windows with the NVDMPPMS (NVDM Presentation Manager Server) command, or by using a response file with the NVDMINST command.

Both methods require the recommended directory structure to be present. The recommended 'CID structure in 1.4.2 on page 29' is used with separate IMG, CSD, RSP and LOG directories for storing the various product and ServicePak, FixPak and Corrective Service Diskette (CSD) images, product response files, and log files under the appropriate subdirectories.

NetView DM/2 defaults SHAREA to store product images and response files and SHAREB to store log files. The IBM NVDM2.INI reflects the two directory areas. In our scenarios, the default values were overridden to \CID and \CID\LOG, respectively.

In our scenarios, we installed the Extended Package of NetView DM/2 on the server using the images previously loaded into the NVDM/2 directory

structure. (D:\CID\IMG\NVDMV21) (Replace 'D:' with whichever drive you are using). For more information about setting up NetView DM/2, please refer to the *NetView Distribution Manager/2 Installation and Customization Guide*. It is advisable to :

- Backup all essential files, for example CONFIG.SYS and STARTUP.CMD, because both these files will be changed by the NetView DM/2 installation.
- Verify that DB/2 has been started and that you are logged on to it with a user ID that has SYSADM authority (the NVDM/2 database will be built during the install). Refer to "Database 2 for OS/2 Considerations" on page 343 on how to customize DB2.

10.3.1 Installing NVDM/2 Server through a Response File

This step shows the non-interactive-way of installing a NetView DM/2 server. To install NVDM/2 server without user action, create a response file as displayed below:

```
//  
// Command Line Values  
//  
BootDrive      =C                // Drive where CONFIG.SYS resides  
SourceDir      =D:\CID\IMG\NVDM2V21  
TargetDir      =D:\NVDM2V21  
LogFile        =C                // Install Stand-Alone Server  
CDM            =C                // Install LDU Distributor  
LDUDistributor=Yes              // Install LDU Change File Builder  
LDUManager     =Yes  
//  
// Configuration Values  
//  
ServerName     =ITSCNVDM        // Length max. 8 characters  
FileServiceDir=D:\FSDATA  
SharedDirA     =D:\CID  
SharedDirB     =D:\CID\LOG  
MaxShrFiles    =1500           // Up to 10,000 is supported  
MaxClients     =10             // Up to 1,000 is supported  
MaxRequests    =8              // Range is from 1 to 8  
ErrorLogFile   =C:\OS2\INSTALL\NVDMINST.ERR  
AdapterNum     =0              // Either Adapter 0 or 1
```

Figure 75. NVDM/2 Server Response File Used by the NVDMINST Command

10.3.2 Installing NVDM/2 Server Interactively

Start the installation process from the diskette images stored at the server.
Proceed as follows:

1. At the server, execute the following command:

```
D:  
CD \CID\IMG\NVDM2V21  
NVDMFMS
```

The 'NVDM/2 Installation' screen is displayed; press **Enter** or select **OK**.



Figure 76. NVMD/2 Installation Screen

2. At the Installation Initialisation window, press **Enter** or click on **Continue**.

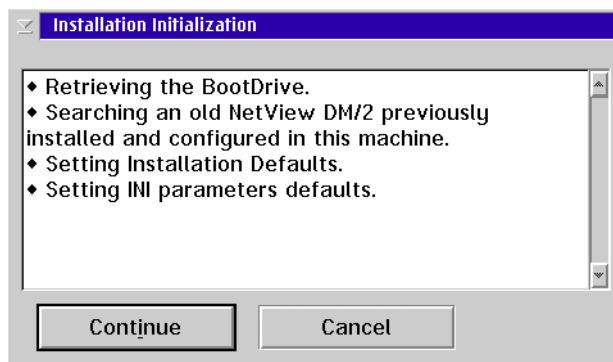


Figure 77. Startup of Installation

The installation program will search for existing NetView DM/2 installations on your workstation. If there is no previous installation, default settings will be displayed as shown in Figure 78 on page 348.

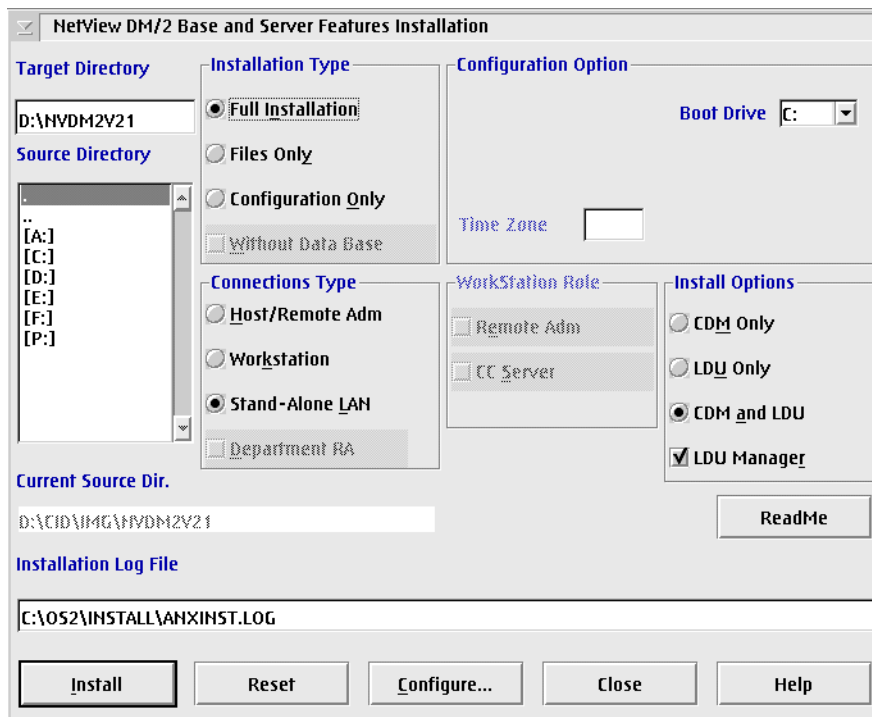


Figure 78. NetView DM/2 Base and Server Features Installation Window

- The 'NetView DM/2 Base and Server Features Installation' window displays defaults which have been set for your environment. The installation parameters you define at this point will be stored in the IBMNVDM2.INI file in the specified target directory.

Target Directory	D:\NVDM2V21
Installation Type	Full Installation
Configuration Option	Boot Drive C:
Connection Type	Stand-Alone LAN
Install Options	CDM and LDU plus LDU Manager

- Select **Configure...** to define the next set of parameters. The NetView DM/2 Base and Server Features (Configure) windows is displayed as shown in Figure 79 on page 349.

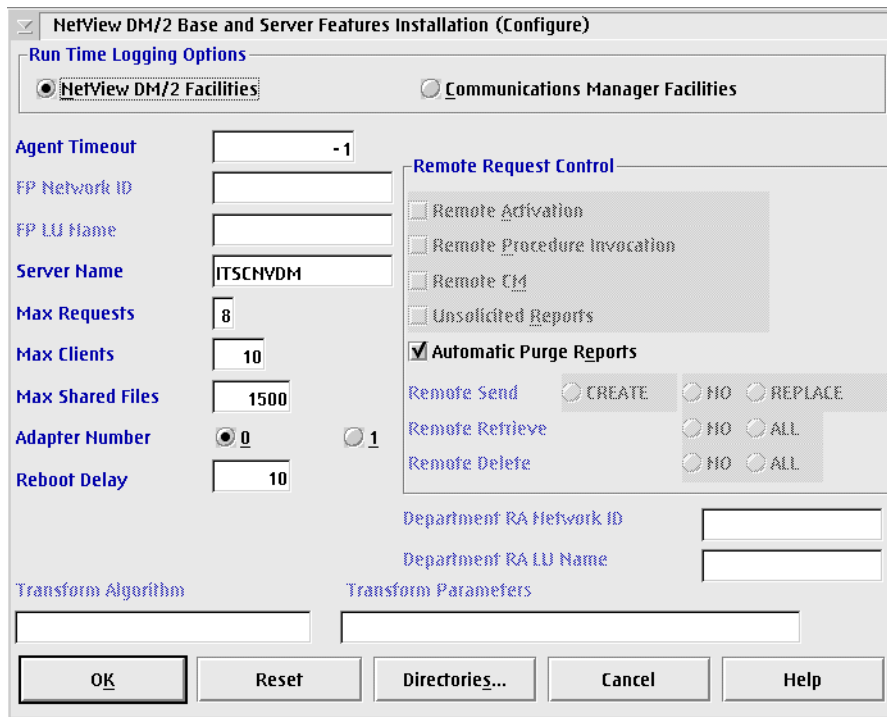


Figure 79. NVDM/2 Base and Server Features Installation (Configure) Window

The following information must be provided:

Run Time Logging Options	NetView DM/2 Facilities
Agent Timeout	-1
Server Name	<server name>
MaxClient	<10>
Adapter Number	<adapter you have defined for NetBIOS>

- After all information has been provided, select **Directories ...** to provide CID directory structure information:

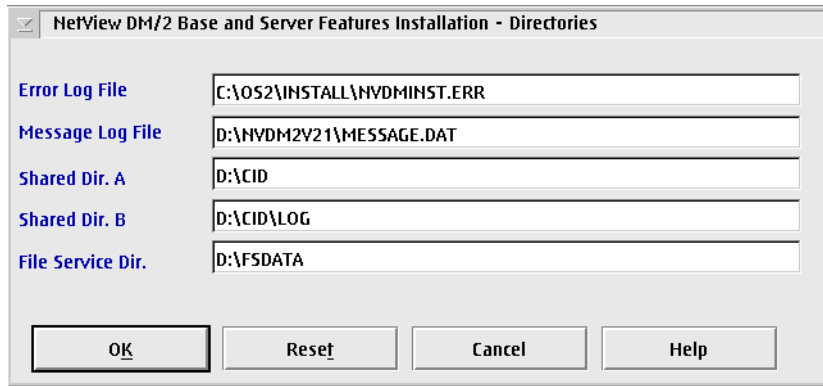


Figure 80. NVDM/2 Base and Server Features Installation - Directories Window

Select directories and set as required:

Error Log File	C:\OS2\INSTALL\NVDMINST.ERR
Message Lof File	D:\NVDM2V21\MESSAGE.DAT
Shared Dir. A	D:\CID
Shared Dir. B	D:\CID\LOG
File Service Dir.	D:\FSDATA

6. Select **OK** to go to the configuration menu and **OK** again to go to the main menu. From there, start the installation by selecting **Install**.
7. During the installation process, a progress indicator will appear. If the installation is complete, select **Continue** to end the installation program as shown in Figure 81 on page 351.

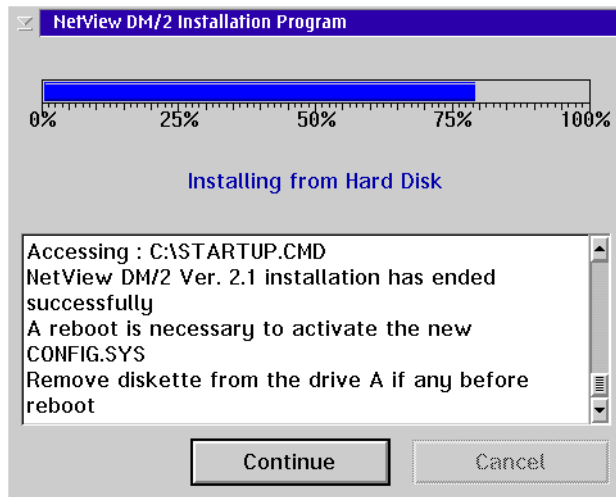


Figure 81. NetView DM/2 Installation Progress Window

8. Modify the `\NVDM2V21\IBMNVDM2.INI` file to give the `SharedDirA` directory read only access:
`SharedDirA=D:\CID,R`
9. Check if the NetBIOS Resource definitions are adequate to suit the requirement of your environment. For more information, please refer to “NetBIOS Resource Considerations” on page 341.
10. Check that the `STARTUP.COMD` is correct as indicated in “Modifying `STARTUP.COMD` Before Installing Netview DM/2” on page 345.
11. Reboot your system.

To check if NetView DM/2 has started successfully, use the following command from an OS/2 command line:

```
CDM STATUS
```

All components should be active and the returned message should be as follows:

```
The CDM CHANGE CONTROLLER status is active.  
The CDM AGENT status is active.
```

10.4 NetView DM/2 Databases

During installation of NetView DM/2, a database will be created at either a local Database 2 server or remote Database 2 server. In our scenario, we

decided on a local database. Using the Database Director, you can easily access the database for individual reports. For example, you can run a query against the NVDM2V21 database to find out what software packages have been distributed to which clients or which clients have gotten ServicePaks, which ones not, and so forth.

The following figure shows the DB2 Database Director. The databases are defined at the server: NVDM2V21 and SAMPLE.

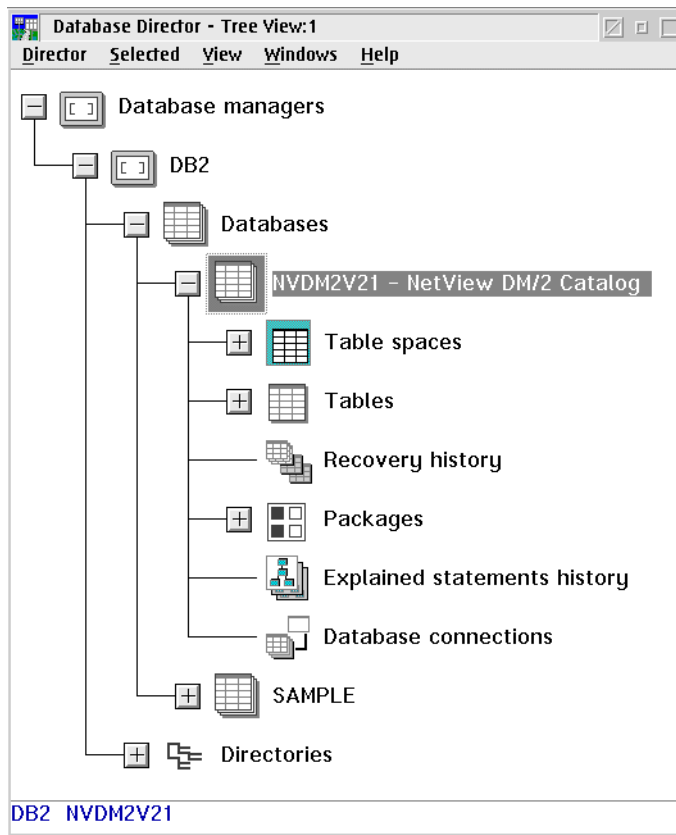


Figure 82. DB2 Database Director

10.5 Global Names, Objects and Catalogs

Any software to be distributed from a NetView DM/2 server must be defined in the CDM catalog before it can be distributed. In order to control how the software or data is being installed on the target workstation, you must build *change files*.

You can create a change file by interactively using the CDM dialogs window or by using the `CDM BUILD` command from an OS/2 command line or OS/2 batch file (.CMD).

10.5.1 Global Names

Global Names are used to uniquely identify an object.

They consist of 2 - 10 tokens, each containing up to 16 alphanumeric or special characters. A period is used to separate the tokens, and the total length of the global name must not exceed 64 characters, including the periods.

10.5.2 Objects

Objects are used by NetView DM/2 to distribute data and install software. An object can contain any kind of software or data. Information about an object is stored in the catalog. Normally, you identify an object by:

- Global name
- Local filename of the physical file that contains the object's data
- Type

10.5.3 Catalog

The catalog is the CC server database, it contains:

- Information about the CC domain and its CC clients
- Software available and distribution status in the CC domain
- Tracking of the change management process

Once the CC server has been set up, the catalog must be prepared in order to be able to maintain the information and software for distribution and installation. All the items (objects) in the catalog are identified by global names.

10.5.4 Change Files

This file describes how an object is to be installed on the target workstation.

Objects that are in the catalog can be used to:

- Install new software
- Refresh previously installed software with a new release
- Upgrade previously installed software with a new level

- Apply a CSD
- Apply a fix

These objects are called change files.

The Global name includes up to seven tokens containing the component name, followed by the change type, such as REF (refresh), UPD (update), or FIX. The global names used in the scenarios in this book are examples only. It is recommended that you define your own global naming standards in order to make the change management process easy to follow.

10.6 Response files for NetView DM/2

The response files for NetView DM/2 are flat ASCII files containing general and/or specific installation and configuration keywords and parameters to facilitate unattended (where possible) distribution of software. See “Creating Response Files” on page 181, for response files being used in our scenarios.

10.7 Preparing a Client Workstation

The next section illustrates client-specific activities to be done to make a client ready for software distribution.

10.7.1 Creation of Boot Diskettes

In order to install a pristine client, a workstation must be booted using three diskettes to load a minimal OS/2 Warp V4, NetView DM/2 CC Client and network drivers to connect to the CC server. The following utilities are used to create these diskettes:

- `SEDISK` Creates the three OS/2 Warp Version 4 diskettes
- `THINLAPS` Installs a minimal LAN support on the boot diskettes
- `NVDMBDSK` Adds the NetView DM/2 client to the boot diskettes

1. Create three OS/2 system boot diskettes

`SEDISK` was copied to the CC server in “Loading OS/2 CID Utilities to the Code Server” on page 76.

Assuming the `\DISK_7\CID` macro file has been unpacked in previous steps, issue the following command:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A:
```

PCMCIA Support

If you are are creating boot diskettes for ThinkPads, add the `/P:` parameter to the `SEDISK` command and pass the line number in `PCMCIA.TBL` that describes the ThinkPad you want boot diskettes for. Note that `PCMCIA.TBL` is OS/2-version dependent and can be found in `\OS2\INSTALL` or in `\DISK_3\BUNDLE` in case of OS/2 Warp 4.

For example, to install OS/2 Warp 4 on a ThinkPad 760ED (Line 36 in `PCMCIA.TBL` from OS/2 Warp 4 represents this ThinkPad), issue:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A: /P:36
```

Insert a formatted diskette into drive A: and press **Enter** when prompted. Insert another diskette when prompted and press **Enter**. Label the first diskette "NetView DM/2 boot diskette 0", the second diskette "Netview DM/2 boot diskette 1". `SEDISK` will display a message when complete. Remove the third diskette and label it "NetView DM/2 Boot Diskette 2".

2. Install LAN transport system to the last boot diskette.

Reinsert the third boot diskette 'NetView DM/2 Boot Diskette 2' into drive A:, and issue the following command :

```
D:\CID\IMG\MPTS\THINLAPS D:\CID\IMG\MPTS A: IBMTOK.NIF
```

This adds LAN Adapter and Protocol Support to the third diskette. `THINLAPS` adds the NetBIOS protocol drivers and the token-ring adapter driver and prompts you to insert boot Diskette 2 in order to update the `CONFIG.SYS` file.

When complete, you will receive a `ThinLAPS` completed successfully message.

3. `NVDMBDSK` installs a minimal NVDM/2 CC client on to the third diskette. `NVDMBDSK.EXE` is located in the `<NetView DM/2 install directory>\BIN` directory on the installed CC server.

Updated NVDMBDSK Files

Find an updated version of the `NVDMBDSK` utility on the enclosed CD-ROM. This `NVDMBDSK` utility has been changed to prompt the user with a pop-up message requesting to insert the boot diskette with the `CONFIG.SYS` file before it continues the process.

If the `NVDMBDSK.EXE` has not been updated to cater for the three boot diskettes, which is required for OS/2 Warp 4, the following steps are required.

1. Copy the CONFIG.SYS file from boot Diskette 1 to a temporary directory on the hard disk and then to boot Diskette 2.
2. Leave boot diskette 2 in the A: drive and issue the following command from an OS/2 command line:

```
D:\NVDM2V21\BIN\NVDMBDSK
```

The NVDM/2 'Boot Diskette Update' window appears showing the following parameters

Target Environment	Operating System of the client to be installed.
Target Drive	Drive where the diskette is inserted.
Server Name	Name of the CC server to which the CC Client will be connected.
	If you enter a "?", you will be prompted to enter the name of the server during the boot sequence.
Client Name	CC Client Name of the pristine workstation.
	Entering a "?" will prompt you to enter a client name.
Attach Timeout	This specifies how long the CDM agent on the client will wait after attempting to attach to the CC server.
	-1 disables the timeout
Receive Timeout	This specifies how long the CDM agent on the diskette initiated will wait for a reply when accessing the CC server shared disk area.
	If this time is exceeded, the CDM agent assumes the CC server is inactive and shuts itself down.
	-1 disables the timeout
Request Timeout	This value specifies how long the CDM agent on the client will wait for a request before terminating.
	-1 disables the timeout

0 terminates immediately; no further requests are pending.

Adapter Number Adapter number to be used when connecting to the LAN.

In our scenarios, the following values were entered:

Target Environment OS/2

Target Drive A:

Server Name ?

Client Name ?

Attach Timeout -1

Receive Timeout -1

Request Timeout -1

Adapter Number 0

Note: Whatever name is used as the client name must be defined on the CC server to enable connection, whether it is defined when creating the boot diskettes (above) or when prompted for the name during the boot sequence (By using the '?' for Server and/or Client name.

Click on **OK**, the following message should appear:

The diskette has been updated correctly.

Select **EXIT** to exit.

Select **YES** to confirm Exit.

3. Copy the CONFIG.SYS to the temporary directory on the hard disk again, remove boot Diskette 2, insert boot Diskette 1, and copy the CONFIG.SYS to boot Diskette 1.

After the completion of SEDISK, THINLAPS, and NVDMBDSK, the additions made to the CONFIG.SYS file on NETVIEW DM/2 boot Diskette 1 are displayed in Figure 83 on page 358.

```

***NVDM Additions are highlighted***

buffers=32
iopl=yes
memman=noswap
protshell=sysinst1.exe
SET OS2_SHELL=\IBMNVDM2\BIN\ANXPULAG.EXE
diskcache=64,LW
protectonly=yes
libpath=.;\os2\dll;\IBMNVDM2\DLL;Z:\DLL;Z:\REXX;
ifs=hpfs.ifs /c:64
.
.
set
path=\;\os2;\os2\system;\os2\install;\IBMNVDM2\BIN;Z:\BIN;Z:\REXX;
set dpath=\;\os2;\os2\system;\os2\install;\IBMNVDM2;Z:\;Z:\REXX;
set keys=on
.
.

rem ***Start of ThinLAPS additions***
device = lanmsgdd.os2
device = protman.os2
device = netbeui.os2
device = netbios.os2
device = IBMTOK.OS2
run = netbind.exe
run = lanmsgex.exe
DEVICE=\IBMNVDM2\BIN\ANXACAIP.SYS
DEVICE=\IBMNVDM2\BIN\ANXIFPID.SYS
DEVICE=\IBMNVDM2\BIN\ANXIFCOM.SYS
IFS=\IBMNVDM2\BIN\ANXIFCOM.IFS

```

Figure 83. NVDM/2 CONFIG.SYS Modifications

10.8 REXX Enablement

This section describes how REXX is activated in the pristine environment (for example, while booted from the CC client boot diskettes).

Since the pristine environment is diskette based, there are no subdirectories at the CC client that can be used to store the REXX files. Therefore, the REXX files are stored on the CC server, and a program is invoked at the CC client to activate REXX.

10.8.1 Setup at the CC Server

Assuming the recommended directory structure has been created, proceed with the following steps:

1. At the CC server, open an OS/2 window and execute the following commands:

```
D:  
CD \NVD2M2V21\PCACODE  
MD REXX
```

The D:\NVD2M2V21\PCACODE directory becomes the redirected drive Z: on the CC client when it boots up and connects to the CC server as a pristine client.

2. To copy the OS/2 Warp 4 REXX executables, issue the following command from an OS/2 window:

```
UNPACK D:\CID\IMG\OS2WARP4\DISK_4\REXX D:\NVD2M2V21\PCACODE\REXX
```

This command unpacks the compressed file named REXX, which includes the following thirty files:

```
D:\NVD2M2V21\PCACODE\REXX\OREXX.DLL  
D:\NVD2M2V21\PCACODE\REXX\REXX.DLL  
D:\NVD2M2V21\PCACODE\REXX\REXX.IMG  
D:\NVD2M2V21\PCACODE\REXX\REXXCRT.DLL  
D:\NVD2M2V21\PCACODE\REXX\OREXXSOM.DLL  
D:\NVD2M2V21\PCACODE\REXX\SOMSEC.DLL  
D:\NVD2M2V21\PCACODE\REXX\REXXAPI.DLL  
D:\NVD2M2V21\PCACODE\REXX\OREXUTIL.DLL  
D:\NVD2M2V21\PCACODE\REXX\OREXXSC.DLL  
D:\NVD2M2V21\PCACODE\REXX\REXXUTIL.DLL  
D:\NVD2M2V21\PCACODE\REXX\REXX.IR  
D:\NVD2M2V21\PCACODE\REXX\OREX.MSG  
D:\NVD2M2V21\PCACODE\REXX\RXQUEUE.EXE  
D:\NVD2M2V21\PCACODE\REXX\RXSUBCOM.EXE  
D:\NVD2M2V21\PCACODE\REXX\OREXH.MSG  
D:\NVD2M2V21\PCACODE\REXX\REXH.MSG  
D:\NVD2M2V21\PCACODE\REXX\WPSINIT.WPS  
D:\NVD2M2V21\PCACODE\REXX\REXXINIT.DLL  
D:\NVD2M2V21\PCACODE\REXX\REXXTRY.CMD  
D:\NVD2M2V21\PCACODE\REXX\WPCLS.IMP  
D:\NVD2M2V21\PCACODE\REXX\WPCONST.CMD  
D:\NVD2M2V21\PCACODE\REXX\WPREXX.IMP  
D:\NVD2M2V21\PCACODE\REXX\OREXXWPS.DLL  
D:\NVD2M2V21\PCACODE\REXX\SWITCHRX.CMD
```

```
D:\NVD2M2V21\PCACODE\REXX\WPFIND.CMD
D:\NVD2M2V21\PCACODE\REXX\REXXC.EXE
D:\NVD2M2V21\PCACODE\REXX\REX.MSG
D:\NVD2M2V21\PCACODE\REXX\DLFCLASS.CMD
D:\NVD2M2V21\PCACODE\REXX\WPSINST.CMD
D:\NVD2M2V21\PCACODE\REXX\WPSYSOBJ.CMD
```

This step copy the files necessary for OS/2 Warp 4 REXX function to the appropriate location for access by NetView DM/2 clients.

3. Copy CMD.EXE from the disk images. Issue the following command from an OS/2 window:

```
COPY D:\CID\IMG\OS2WARP4\DISK_1\CMD.EXE D:\NVD2M2V21\PCACODE\REXX
```

The OS/2 command processor, CMD.EXE, is required in an available drive for batch files to execute. It does not require explicit execution.

4. Copy SRVREXX.EXE from the OS/2 Warp 4 CID directory. Issue the following command:

```
COPY C:\CID\LOCINSTU\SRVREXX.EXE D:\NVD2M2V21\PCACODE\REXX
```

assuming OS/2 Warp 4 is installed on the C: drive. SRVREXX.EXE is the program that loads REXX from the server onto a client workstation.

5. Modify the client's CONFIG.SYS to permit REXX function. Insert NetView DM/2 boot Diskette 1 into the diskette drive and enter the following command:

```
EPM A:\CONFIG.SYS
```

and add the following text to the end of the LIBPATH, PATH and DPATH statements:

```
Z:\REXX;
```

6. Create the REXX command file. Issue the following command from an OS/2 window:

```
EPM D:\NVD2M2V21\PCACODE\REXX\REXSTART.CMD
```

and enter the following information:

```
@DETACH Z:\REXX\SRVREXX.EXE
```

The created REXSTART.CMD batch file initiates the REXX function at the client. DETACH starts and simultaneously detaches a program from the command processor so it can be run in the background.

7. Create a REXSTART change file to enable REXX in the pristine environment as described in the next section.

10.9 Creating Change Files

Any software meant for distribution must be cataloged in the CDM catalog. To catalog software, a change file must be created for each software package to be distributed.

We suggest building 'Change File Profiles' instead of using the panels provided by NetView DM/2 to create change files. This makes it easier to edit/change the profile to re/build, and it is easier to back up the profiles. Store the profiles in D:\CID\PRF.

More syntax information of CID-related utilities are provided in Chapter Chapter 4., 'Utilities to Ease Remote Installation' on page 75 .

10.9.1 The REXX Start Profile

The following profile enables REXX at the pristine client by invoking the REXSTART.COMM command that executes the SRVREXX.EXE program to initialize REXX.

```
TargetDir=C:\

Section Catalog
Begin
  ObjectType   = Software
  GlobalName   = ITSCAUS.REXX.PRISTINE.REF.4
  Description   = "Procedure to load REXX on Pristine Workstation"
End

Section Install
Begin
  Program      = Z:\REXX\REXSTART.COMM
End
```

Figure 84. REXX Start Change Profile

Note that REXSTART.COMM was created in list item 7 on page 360.

10.9.2 The FDISK and Format Profile

The `FDISK` command to partition the drive can be run in one of many ways, such as running `FDISK` from the command line after booting from diskettes, running a command file initiated from the diskettes, or using the software distribution manager (such as NVDM/2) to initiate the process.

The following profile executes a command file called PREPWS.CMD, an example of which can be found in the `FDISK` section, along with more information about `FDISK`.

Store the profile in the `\CID\PRF` directory as `FDISK.PRO`.

```
TargetDir = "C:\"

Section Catalog
Begin
    ObjectType = SOFTWARE
    GlobalName = ITSCAUS.OS2WARP4.PREPWS.REF.1
    Description = Automated Partitioning for OS/2 PC's
End

Section Install
Begin
    Program = SA:\IMG\OS2WARP4\CMD.EXE
    Pargs = "/C DISKPRP.CMD"
    WorkingDir = SA:\EXE
End
```

Figure 85. FDISK Change Profile

Note: Make sure `CMD.EXE` exists in the `\IMG\OS2WARP4` directory.

10.9.3 The OS/2 Warp 4 Profile

The following figure shows the OS/2 Warp 4 change profile that resides in the `\CID\PRF` directory as `OS2WARP4.PRO`.

```

TargetDir=C:\

Section Catalog
Begin
  ObjectType   = Software
  GlobalName   = ITSCAUS.OS2Warp4.INST.REF.4
  Description  = "Install Procedure for OS/2 Warp 4 through SEINST"
End

Section Install
Begin
  Program      = SA:\IMG\OS2Warp4\SEINST.EXE
  Parms        = /S:${SourceDir} /B:C: /R:${responsefile}
                                     /l1:${logfile1} /T:A:\

  SourceDir    = SA:\IMG\OS2Warp4
  ResponseFile = SA:\RSP\OS2Warp4\$(WorkStatName).RSP
  Logfile1     = SB:\OS2Warp4\$(WorkStatName).LOG
End

```

Figure 86. OS/2 Warp 4 Change Profile

Note: The `Parms` line must be a single line.

10.9.4 The OS/2 Warp 4 Maintenance Partition Profile

The following figure displays the OS/2 Warp 4 Maintenance Partition (SEMAINT) profile that resides in the \CID\PRF directory as SEMAINT.PRO.

```

TargetDir=E:\SERVICE

Section Catalog
Begin
  ObjectType=Software
  GlobalName=ITSCAUS.OSWARP4.MAINT.REF.4
  Description="SEMAINT Procedure for OS/2 Warp 4"
End

Section Install
Begin
  Program = SA:\IMG\OS2WARP4\SEMAINT.EXE
  Pams= /S:${SourceDir} /B:E: /l1:${logfile1} /T:${TargetDir}
  SourceDir= SA:\IMG\OS2WARP4
  LogFile1 = SB:\LOG\OS2WARP4\$(WorkStatName).MNT
End

```

Figure 87. OS/2 Warp 4 Maintenance Partition (SEMAIN) Profile

10.9.5 The MPTS Install Profile

The MPTS Install Profile can be created and used in two different types of environments:

- The profile that runs in a pristine (non-Presentation Manager) environment

```

TargetDir=C:\

Section Catalog
Begin
  ObjectType = Software
  GlobalName = ITSCAUS.MPTS.INST.REF.5
  Description = "Install Procedure for MPTS WR08415"
End

Section Install
Begin
  Program      = $(SourceDir)\MPTS.EXE
  Pargs        = /S:$(SourceDir) /T:$(TargetDir) /TU:$(TargetDir)
                /R:$(responsefile) /L1:$(logfile1) /E:MAINT
  SourceDir    = SA:\IMG\MPTS
  ResponseFile = SA:\RSP\MPTS\$(workStatname).RSP
  Logfile1     = SB:\MPTS\$(WorkStatName).LOG
End

```

Figure 88. MPTS Maintenance Environment Change Profile

Note: The `Pargs` line must be a single line.

- The profile that runs in a production environment (Presentation Manager available). In this case additional objects will be created for you in the System Setup folder, for instance:
 - MPTS
 - DDNS Configuration
 - DHCP Monitor

If you do not need these additional objects, you do not need to send this change file to the the remote workstations.

```

TargetDir=C:\

Section Catalog
Begin
  ObjectType = Software
  GlobalName = ITSCAUS.MPTSPM.INST.REF.5
  Description = "Install Procedure for MPTS WR08415 through PM"
End

Section Install
Begin
  Program      = $(SourceDir)\MPTS.EXE
  Params       = /S:$(SourceDir) /T:$(TargetDir) /TU:$(TargetDir)
                /R:$(responsefile) /L1:$(logfile1)

  SourceDir    = SA:\IMG\MPTS
  ResponseFile = SA:\RSP\MPTS\$(workStatname).RSP
  Logfile1     = SB:\MPTS\$(WorkStatName).LOG
End

```

Figure 89. MPTS Production Environment Change Profile

Note: The `Params` line must be one single line.

10.9.6 The NVDM/2 CC OS/2 Client Install Profile

The NetView DM/2 Software Distribution agent can be installed remotely in three different ways:

- The agent that will be installed with the OS/2 base install immediately after MPTS (all components installed as a corequisite group).

```

TargetDir          = C:\IBMNVDM2

Section Catalog
Begin
  ObjectType       = SOFTWARE
  GlobalName       = ITSCAUS.NVDMCLI.INST.REF.2.1
  Description      = "Install Procedure for NVDM/2 Client"
End

Section Install
Begin
  Program          = SA:\IMG\NVDM2V21\NVDMCLT.EXE
  Parms            = /B:C /R:${ResponseFile} /S:${SourceDir}
                  /T:${TargetDir} /L:${LogFile1}

  ResponseFile     = SA:\RSP\NVDM2V21\$(WorkStatName).RSP
  SourceDir        = SA:\IMG\NVDM2V21
  LogFile1         = SB:\NVDM2V21\$(WorkStatName).LOG
  PhaseEnd=Yes
End

```

Figure 90. NVDM/2 Client Change Profile Part of a Corequisite Group Install

Note: The `Parms` line must be a single line.

- The agent that will be installed on a so-called *seed* environment directly after `THINLAPS` is installed and configured in the maintenance partition of the remote system.

This procedure is optional and only useful if you want to have only one copy of the NetView DM/2 agent code on your target system. The production environment and maintenance partition would share this one copy. Usually this is not a useful approach since in most cases, you want to two different agents running independently from each other if you decided to install maintenance partitions on your target systems. The next bullet item takes care of this kind of scenario.

```

TargetDir          = E:\IBMNVDM2

Section Catalog
Begin
  ObjectType       = SOFTWARE
  GlobalName       = ITSCAUS.NVDMCLI.MAINT.REF.2.1
  Description      = "Install Procedure for NVDM/2 Client Maintenance"
End

Section Install
Begin
  Program          = SA:\IMG\NVDM2V21\NVDMCLT.EXE
  Parms            = /m /B:C /R:$(ResponseFile) /S:$(SourceDir)
                   /T:$(TargetDir) /L:$(LogFile1)

  ResponseFile     = SA:\RSP\NVDM2V21\$(WorkStatName).RSP
  SourceDir        = SA:\IMG\NVDM2V21
  LogFile1         = SB:\NVDM2V21\$(WorkStatName).LOG
  PhaseEnd         = Yes
End

```

Figure 91. NVDM/2 Change Profile: Maintenance Partition Agent

Note: The `Parms` line must be a single line. The maintenance partition is assumed to be drive `E:`.

- The agent that will be installed when the production environment has been installed and when the maintenance partition is created on the workstation using `SEMMAINT` and `THINLAPS`. The agent is necessary in order to have independent NVDM/2 clients running at the remote client, for instance:
 - OS/2 Production Environment
 - OS/2 Maintenance Partition

or the maintenance environment installation. The REXX CMD file in Figure 93 on page 370 should allow two agents to coexist.


```

TargetDir          = E:\

Section Catalog
Begin
  ObjectType       = SOFTWARE
  GlobalName       = ITSCAUS.NVDMCCC.MAINT.REF.2.1
  Description      = 'Software Distribution Agent Maintenance Partition'
end

Section Install
Begin
  Program          = $(SourceDir)\EXE\SD_MAINT.CMD
  Pargs            = $(SourceDir)\IMG\SD_MAINT
  SourceDir        = SA:
End

```

Figure 92. NVDM/2 Change Profile: Maintenance Partition Agent (Alternative)

Note: The `Pargs` line must be a single line. The maintenance partition is assumed to be drive E:.

The install process of the NetView DM/2 client does not allow two copies to exist on the client. One copy would be overwritten with either the OS/2 production or the maintenance environment installation.

The `\CID\IMG\SD_MAINT` directory contains a copy of the NetView DM/2 agent from an already installed maintenance partition-NetView DM/2 client. Do not use the NetView DM/2 agent that is installed on boot diskettes since this one would not work out of a maintenance partition. NetView DM/2 has two different agents: one for boot diskettes and one for the agent installed on a maintenance partition.

```

/* REXX CMD file that copies the NVDM/2 software Distribution */
/* Agent to a maintenance Partition and creates a IBMNVDM/2.INI file */
/* file with the correct workstation name and path statements) */

arg source_dir

Ini_file = 'C:\IBMNVD2.INI'
linenum=0

'XCOPY 'source_dir'\IBMNVD2\*. * E:\IBMNVD2\*. * /s /e'
'XCOPY 'source_dir'\CONFIG.SYS E:\*. *'

Do While lines(Ini_file)
  line.linenum = Linein(Ini_file);
  PARSE VAR line.linenum p1 p2 p3 p4
  if SUBSTR(p3,1,11) = 'C:\IBMNVD2' then p3= OVERLAY('E:\IBMNVD2',p3)
    if linenum = 0
      then 'ECHO 'p1 p2 p3 p4 '> E:\IBMNVD2\IBMNVD2.INI'
      else 'ECHO 'p1 p2 p3 p4 '>>E:\IBMNVD2\IBMNVD2.INI'
    linenum=linenum+1
End

exit 0

```

Figure 93. REXX Procedure that Copies the Second Agent (SD_MAINT.COM)

Note: The `if SUBSTR(p3,1,11)` line must be a single line. The maintenance partition is assumed to be drive E:

10.9.7 The Feature Installer V1.2.1 (FISSETUP) Change Profile

We recommend installing the latest Feature Installer version available from IBM's Software Choice home page on the Internet at:

<http://www.software.ibm.com/os/warp/swchioce>

At the time this redbook was written, Version 1.2.1 was available. Install this version to assure coexistence between ASD (Alternate Software Delivery), OS/2 Warp 4 Corrective Service Facility and other installations through Feature Installer. The following change profile shows how to install the new version.

```

TargetDir=C:

Section Catalog
Begin
  ObjectType      = Software
  GlobalName      = ITSCAUS.FI.INST.REF.1.2
  Description     = "Install Procedure for Feature Installer (FI)"
End

Section Install
Begin
  Program         = SA:\IMG\FISETUP\FISETUP.EXE
  Parns           = /B:$(TargetDir) /S:$(SourceDir) /L1:$(LogFile1) /NN
  SourceDir       = SA:\IMG\FISETUP
  LogFile1       = SB:\FISETUP\$(WorkStatName).FI
End

```

Figure 94. FISETUP Feature Installer Change Profile

10.9.8 The Command Line Interface Feature Installer Change Profile

To distribute OS/2 Warp 4 features such as Java, Developer Extensions or BonusPak components, the Feature Installer needs to be executed when the base operating system along with MPTS and the NVDM/2 clients have installed successfully and the system has rebooted.

```

TargetDir=C:

Section Catalog
Begin
  ObjectType      = Software
  GlobalName      = ITSCAUS.CLIFI.INST.REF.4
  Description     = "Install Procedure for OS/2 Warp 4 Part 2 through
CLIFI"
End

Section Install
Begin
  Program         = SA:\IMG\OS2Warp4\CLIFI.EXE
  Parms           = /A:C /F:C:\OS2\INSTALL /B:C: /S:${SourceDir}
                  /R:C:\OS2\INSTALL\FIBASE.RSP /R2:${ResponseFile}
                  /L1:${LogFile1} /L2:${LogFile2}

  SourceDir       = SA:\IMG\OS2Warp4\FI
  ResponseFile    = SA:\RSP\OS2Warp4\$(WorkStatName).RSP
  LogFile1        = SB:\OS2Warp4\$(WorkStatName).FI1
  LogFile2        = SB:\OS2Warp4\$(WorkStatName).FI2
End

```

Figure 95. CLIFI Change Profile

Note: The `Parms` line must be a single line. Make sure CLIFI.EXE is copied to the \CID\IMG\OS2WARP4 directory. This can be done from a system on which the latest version of feature installer was applied to. CLIFI.EXE resides on the \OS2\INSTALL directory.

10.9.9 OS/2 Warp 4 FixPak 4 Profile

Assuming you want to distribute XR_M004 (FixPak 4), you need to create a change profile as shown below. Make sure FixPak 4 is copied to the \CID\CSD\XR_M004 directory. Since FixPaks require Kicker Diskettes to start the installation process, copy the diskettes images to the \CID\CSD\KICKER directory.

```

TargetDir = C:\
Section Catalog
Begin
  ObjectType = Software
  GlobalName = ITSCAUS.XR_M004.INST.REF.4
  Description = "Install Procedure for OS/2 Warp 4 ServicePak 4"
End

Section Install

Begin
  Program      = SA:\CSD\KICKER\FSERVICE.EXE
  Params       = /S:${SourceDir} /T:${TargetDir} /R:${Responsefile}
               /L1:${LogFile1}

  ResponseFile = SA:\RSP\XR_M004\RESPONSE.FIL
  SourceDir    = SA:\CSD\XR_M004
  LogFile1     = SB:\LOG\XR_M004\$(WorkStatName).LOG
End

```

Figure 96. OS/2 Warp 4 FixPak 4 (XR_M004) Change Profile

Note: The `Params` line must be a single line.

10.9.10 Display Driver Install for S3 Step 1

In some cases, DSPINSTL must configure the display driver at the remote client. This is true in particular if the display adapter is an S3 adapter. Three steps are needed to install an S3 display driver with higher resolution, higher screen refresh rate and installed monitor. The following profile is in charge of configuring the client with base VGA display driver.

```

TargetDir = C:\
Section Catalog

Begin
  ObjectType = Software
  GlobalName = ITSCAUS.DISPLAY1.INST.REF.4
  Description = "DSPINSTL Step 1"
End

Section Install

Begin
  Program = C:\OS2\INSTALL\DSPINSTL.EXE
  Pams = /S:${SourceDir} /T:C: /PD:C:\OS2\INSTALL\PSVGA32.DSC /U
  SourceDir = SA:\IMG\OS2WARP4
End

```

Figure 97. Display Driver Install (DSPINSTL) Step 1 Change Profile

10.9.11 Display Driver Install for S3 Step 2

The next step of DSPINSTL installs the correct video driver through autodetect with the resolution of 640 by 480 and 256 colors..

```

TargetDir = C:\
Section Catalog

Begin
  ObjectType = Software
  GlobalName = ITSCAUS.DISPLAY2.INST.REF.4
  Description = "DSPINSTL Step 2"
End

Section Install

Begin
  Program = C:\OS2\INSTALL\DSPINSTL.EXE
  Pams = /S:${SourceDir} /T:C: /RES:640x480x256 /AUTO
  SourceDir = SA:\IMG\OS2WARP4
End

```

Figure 98. Display Driver Install (DSPINSTL) Step 2 Change Profile

10.9.12 Display Driver Install for S3 Step 3

The next step is in charge of the correct screen resolution, screen refresh rate and specifying the monitor attached to the workstation.

```
TargetDir = C:\
Section Catalog

Begin
  ObjectType = Software
  GlobalName = ITSCAUS.DISPLAY3.INST.REF.4
  Description = "VCFGCID - DSPINSTL Step 3"
End

Section Install

Begin
  Program = C:\OS2\INSTALL\VCFGCID.CMD
  Parms = /RES:1024x768x256 /RR:72 /MON:137
End
```

Figure 99. Display Driver Install (VCFGCID) Step 3 Change Profile

10.9.13 TCP/IP Change Profile

The following change profile installs the TCP/IP component of OS/2 Warp 4.

```

TargetDir=C:\

Section Catalog
Begin
  ObjectType   = Software
  GlobalName   = ITSCAUS.TCPIP.INST.REF.4
  Description  = "Install Procedure for TCP/IP 4.0"
End

Section Install
Begin
  Program      = SA:\IMG\TCPIP\INSTALL.EXE
  Parms        = /b- /S:${SourceDir} /R:${responsefile}
                                     /L1:${logfile1}

  SourceDir    = SA:\IMG\TCPIP
  ResponseFile = SA:\RSP\TCPIP\$(WorkStatName).RSP
  Logfile1     = SB:\TCPIP\$(WorkStatName).LOG
End

```

Figure 100. TCP/IP Change Profile

Note: The `Parms` line must be a single line.

10.9.14 Network SignOn Coordinator Change Profile

The following change profile installs the Network SignOn Coordinator component of OS/2 Warp 4.


```

TargetDir      = C:\

Section Catalog
Begin
  ObjectType   = SOFTWARE
  GlobalName   = ITSCAUS.NSC.INST.REF.4
  Description  = "Install Procedure for Network SignOn Coordinator/2"
end

Section Install
Begin
  Program      = SA:\IMG\NSC\INSTALL.EXE
  Parms        = /A:I /X /R:$(ResponseFile) /L1:$(LogFile1)
                /L2:$(LogFile2) /S:$(SourceDir)

  SourceDir    = SA:\IMG\NSC
  ResponseFile = SA:\RSP\NSC\$(WorkStatName).RSP
  LogFile1     = SB:\NSC\$(WorkStatName).LOG
  LogFile2     = SB:\NSC\$(WorkStatName).HST
End

```

Figure 101. Network SignOn Coordinator Change Profile

Note: The `Parms` line must be a single line.

10.9.15 OS/2 Peer Services Change Profile

The following change profile installs the OS/2 peer services component of OS/2 Warp 4.

```

TargetDir          = C:\

Section Catalog
Begin
  ObjectType       = SOFTWARE
  GlobalName       = ITSCAUS.OS2PEER.INST.REF.4
  Description      = "Install Procedure for OS/2 Warp 4 Peer
Services"
End

Section Install
Begin
  Program          = SA:\IMG\OS2PEER\PEERRMT.EXE
  Parms            = /CONNECT /R:${ResponseFile} /L1:${LogFile1}
  ResponseFile     = SA:\RSP\OS2PEER\$(WorkStatName).RSP
  LogFile1        = SB:\OS2PEER\$(WorkStatName).LOG
End

```

Figure 102. OS/2 Peer Change Profile

Below are sample change profiles for additional software you may want to install on client machines. These examples assume the applicable product images have been copied to the appropriate directories. For information on response files, refer to "Creating Response Files" on page 181.

10.9.16 Personal Communications for OS2 Warp Profile

The following change profile installs eNetwork Personal Communications for OS/2 Warp.

```

TargetDir=C:\
Section Catalog
Begin
  ObjectType = Software
  GlobalName = ITSCAUS.PCOMOS2.INST.REF.410
  Description = "Base Installation of PCOMOS2 V4.1"
End

Section Install

Begin
  Program = SA:\IMG\PCOMOS2\INSTALL.EXE
  Pargs = /S:${SourceDir} /R:${responsefile} /M:C /L1:${LogFile1}
          /L2:${LogFile2}

  ResponseFile = SA:\RSP\PCOMOS2\$(WorkStatName).RSP
  SourceDir = SA:\IMG\PCOMOS2
  WorkingDir = SA:\IMG\PCOMOS2
  LogFile1 = SB:\PCOMOS2\$(WorkStatName).L01
  LogFile2 = SB:\PCOMOS2\$(WorkStatName).L02
End

```

Figure 103. Personal Communications for OS/2 Change Profile..

Note: The `Pargs` line must be a single line.

10.9.17 Remote Printer Install (RINSTPRN) Change Profile

The following change profile installs printer drivers remotely according to the keywords set in the RPM1 response file.

```

TargetDir = C:\

Section Catalog
Begin
  ObjectType = Software
  GlobalName = ITSCAUS.RMPI.INST.REF.1
  Description = "Base Installation for Remote Printers"
End

Section Install

Begin
  Program = SA:\IMG\RMPI\RINSTPRN.EXE
  Pams = /S:${SourceDir} /DSC:${SourceDir}\PRDESC.LST
        /DRV:${SourceDir}\PRDRV.LST
        /WPR:C:\OS2\MDOS\WINOS2\SYSTEM\CONTROL.INF
        /WDR:C:\OS2\MDOS\WINOS2\SYSTEM\DRVMAP.INF /L1:${LogFile1}
        /R:${ResponseFile}
  ResponseFile = SA:\RSP\RMPI\$(WorkStatName).RSP
  SourceDir = SA:\IMG\WARP4\PMDD_1
  WorkingDir = SA:\IMG\RMPI
  LogFile1 = SB:\LOG\RMPI\$(WorkStatName).LOG
End

```

Figure 104. Remote Printer Install Change Profile

Note: The `Pams` line must be a single line.

10.9.18 Desktop Shuffler Profile

The Desktop Shuffler is in charge of reorganizing the desktop after all subcomponents and networking components have been installed. This change profile must definitely be the last change management request sent to the remote client to ensure a cleaned-up Workplace Shell.

```

TargetDir = C:\

Section Catalog
Begin
  ObjectType = SOFTWARE
  GlobalName = ITSCAUS.ICON.SHUFFLER.REF.1
  Description = "ICON SHUFFLER FOR OS/2 (CLNDESK.EXE)"
End

Section Install

Begin
  Program = SA:\EXE\ISHUFFL.COMD
  Pams = /S:${SourceDir}
  SourceDir = SA:\EXE
  WorkingDir = SA:\EXE
End

```

Figure 105. Desktop Icon Shuffler Change Profile (SHUFFLER.PRO)

10.9.18.1 The ISHUFFL REXX Command File

The following figure displays the ISHUFFL.COMD file executed by the Desktop Shuffler change profile.

```

CLNDESK SHUF_CTL.INI SHUF_KEY.INI

```

Figure 106. Desktop Icon Shuffler REXX Command File

The two input files, SHUF_CTL.INI and SHUF_KEY.INI, are displayed in Appendix A.5, "The Desktop Shuffler Tool" on page 463.

10.10 NetView DM/2 Command Overview

NetView DM/2 can be controlled in two ways: by executing a command from an OS/2 command prompt or through the use of the panels of the dialog interface (the Engine pull-down menu of the CDM Catalog panel). A list of all commands is provided in the *IBM NetView Distribution Manager/2 Version 2.1 User's Guide*, SH19-5048 and in the CDM.INF located in the \NVDM2V21 directory. This can be accessed using the VIEW command.

Some of the global commands that start and stop NetView DM/2 or its subcomponents are:

CDM STATUS	Displays the status of all subcomponents
CDM START	Starts all subcomponents
CDM STOP	Stops all subcomponents immediately (regardless of current requests)
CDM SHUTDOWN	Terminates (all) subcomponents (current requests are completed)

CDM START starts all subcomponents installed on the workstation. For example, on a CC client, only the CDM agent is started because it is the only subcomponent present. CDM STOP and CDM SHUTDOWN stop all installed subcomponents. CDM SHUTDOWN finishes current requests before stopping the subcomponents.

The subcomponents can also be started/stopped individually. The transmission controller can be started (CDM START TRANSM), stopped (CDM STOP TRANSM) and quiesced (CDM QUIESCE). In quiesced status, the transmission controller can perform administrative functions such as queuing of send/receive requests, but the transport layer is not active.

The CDM agent is started by the CDM START AGENT command and stopped by CDM STOP AGENT. The CDM START/STOP MANAGER starts/stops the change controller and transmission controller. CDM SHUTDOWN MANAGER and CDM SHUTDOWN AGENT terminate the CDM manager and CDM agent, respectively. The following shows the usage of the commands and the messages you will get when issuing them.

CDM STOP	The command is in progress. Check the Transmission Controller status later. The STOP request was accepted.
CDM START	The Transmission Controller is starting. Check CDM status later. The START request was accepted.
CDM STATUS	The CDM TRANSMISSION CONTROLLER status is active. The CDM CHANGE CONTROLLER status is active. The CDM AGENT status is active.
CDM QUIESCE TRANSM	The Transmission Controller is quiescing. Check CDM status later.

```
CDM STATUS          The CDM TRANSMISSION CONTROLLER status is quiesced.
                   The CDM CHANGE CONTROLLER status is active.
                   The CDM AGENT status is active.
```

10.10.1 Creating the Software Catalog

Before we can actually initiate remote installations with NetView DM/2, we need to create a CDM catalog. The change profiles created in previous steps are the basis for creating the CDM catalog.

The software catalog (CDM catalog) can be created either through NetView DM/2's graphical user interface or by using the `CDM BUILD` command:

```
CDM BUILD OS2WARP4.PRO FS:OS2WARP4.CHG
CDM BUILD MPTS.PRO     FS:MPTS.CHG
CDM BUILD NVDMCLI.PRO  FS:NVDMCLI.CHG
CDM BUILD FI.PRO       FS:FI.CHG
CDM BUILD CLIFI.PRO    FS:CLIFI.CHG
CDM BUILD XR_M004.PRO  FS:XR_M004.CHG
CDM BUILD OS2PEER.PRO  FS:OS2PEER.CHG
CDM BUILD TCPIP.PRO    FS:TCPIP.CHG
CDM BUILD NSC.PRO      FS:NSC.CHG
CDM BUILD NETFIN.PRO   FS:NETFIN.CHG
CDM BUILD SYSVIEW.PRO  FS:SYSVIEW.CHG
CDM BUILD LDREM.PRO    FS:LDREM.CHG
CDM BUILD REXSTART.PRO FS:REXSTART.CHG
```

Figure 107. Creating the NetView DM/2 Software Catalog

While cataloging the software, the `CDM BUILD` command responses back the following message:

```
BUILD was successful and an entry with Global Name
'ITSCAUS.OS2WARP4.INST.REF.4' was created in the catalog; the Change File
contains only the INSTALL section.
```

Since the software being cataloged is CID-enabled, the change file can only have the `INSTALL` section.

To create the NetView DM/2 software catalog through its graphical user interface, perform the following steps:

1. At the NetView DM/2 - CDM Catalog window, select **File** and then **Build from profile...** as shown in Figure 108.

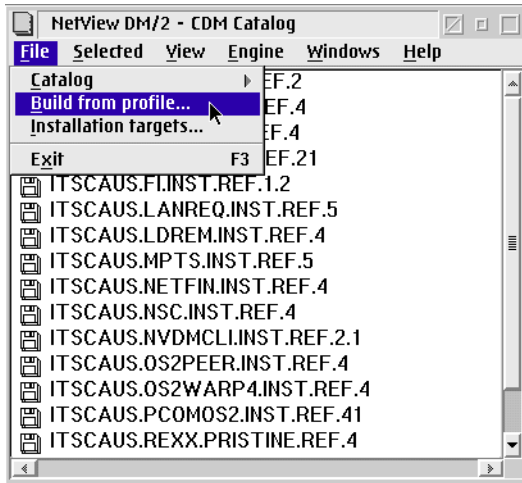


Figure 108. NetView DM/2 - CDM Catalog Window

- At the Build Change File window, as shown in Figure 109, specify the the name of the Change file profile, for example, OS2WARP4.PRO. You may use the **Find...** button to click on the file name through the file list. Also specify the name of the Target file. The path is given already, and the name should be the same as the change profile file name, however, with the file extension of .CHG.

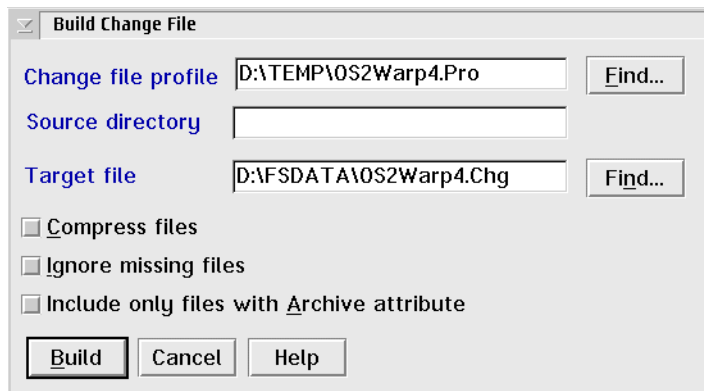


Figure 109. Build Change File Window

- Click on the **Build** button to create the change file.

10.11 Connectivity between CC Server and CC Client

This section describes the procedure to establish connectivity between the CC server and the CC client. To create CC clients at the server, there are two ways:

- Using CDM commands from an OS/2 command line
- Using the graphical user interface

Select the one that suits you most.

10.11.1 Define CC Clients using CDM Commands

If you have a diskette-initiated machine or do not have the NetView DM/2 CC client installed on the workstation, you can temporarily boot the workstation from the NetView DM/2 CC client boot diskettes to connect with the CC server.

The following tasks can also be executed through the CDM dialog. For the syntax of the CDM command line commands, see the online command reference CDM.INF.

At the CC Server:

1. Define CC client.

Issue the following command:

```
CDM ADD_WS <client name>
```

The `ADD_WS` command defines a new client to the server.

`LIST_WS` lists all the clients currently defined.

```
CDM LIST_WS
```

If the client is not started, it will have an inactive status. Notice that the CC server's name is listed as `SERVER`.

2. Start NetView DM/2.

Issue the following command to check the status of NetView DM/2 components:

```
CDM STATUS
```

If any of the NetView DM/2 components are inactive, enter:

```
CDM START
```

All NetView DM/2 components on the CC server should be started automatically by STARTUP.COM. If for some reason they are no longer active, you can issue `CDM START` to start all components installed on the CC server.

`CDM STATUS` displays the status of NetView DM/2 components. Status of the Change Controller and CDM agent should be ACTIVE.

10.11.2 Define CC Clients using the Graphical User Interface

Alternatively, instead of using the command line interface, you can define new workstations through NetView DM/2's graphical user interface. They need to be defined in the CDM CC Domain window. You can open this window by selecting **CC Domain** in the pulldown menu of the **Windows** menu item on the menu bar.

1. At the CDM CC Domain window, select Workstation from the menu bar, and then New... as shown in Figure 110.

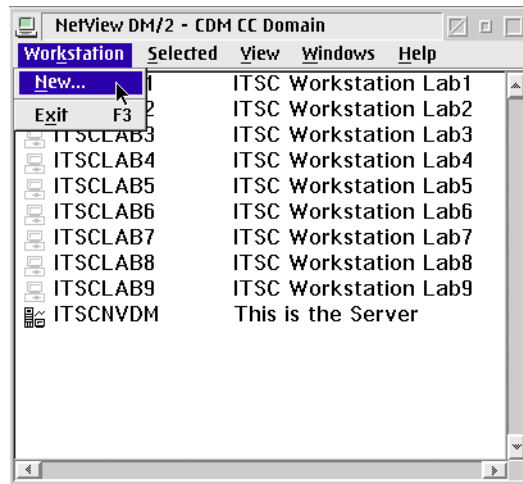


Figure 110. NetView DM/2 - CDM CC Domain Window

The New Workstation window is displayed as shown in Figure 111 on page 387.

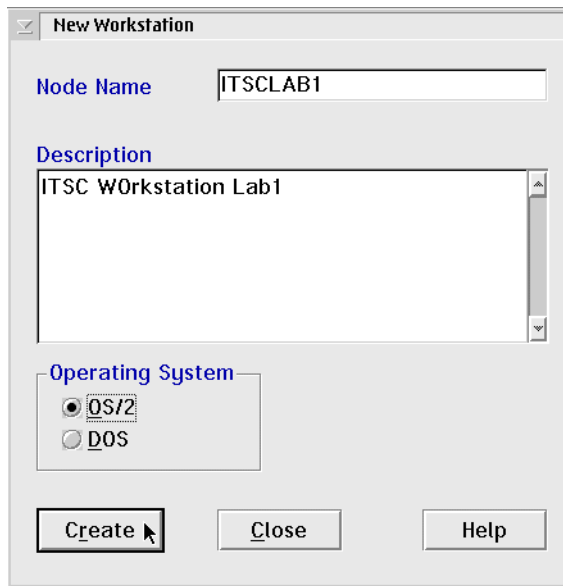


Figure 111. New Workstation Window

2. You need to specify the workstation's node name (up to eight characters long). Optionally, specify the description of the workstation and the operating system the remote client is going to use or is using by clicking on its appropriate radio button. When finished, select **Create** to catalog the new workstation.

10.11.3 Establish Connectivity at the CC Client Workstation

The following activities need to be done at the CC client:

1. Start NetView DM/2.

If NetView DM/2 client code is NOT installed perform the following steps:

- Insert NetView DM/2 boot Diskette 0 into the A: drive and reboot the workstation.
- When prompted, insert NetView DM/2 boot Diskette 1, the boot Diskette 2.

The NetView DM/2 Pristine Client Agent display will appear.

Enter the client and server names:

CLIENT = <client name>

SERVER = <server name>

The CC client name entered here must match the one you defined at the CC server. The server name must be the name specified when NetView DM/2 was installed on the CC server.

- After the client establishes a session with the server, you will receive the following message:

```
ANX1317W CDM Agent Starting. You can remove boot diskette and leave
the workstation unattended
```

Note: If you are going to run the PREPWS.COM (FDISK.PRO) to partition the disk, do not remove the diskette at this point because this command file writes to it.

- When the client is attached to the server, you will receive the following message:

```
ANX1310I the pristine client agent is waiting for CM request message.
```

After all change-management requests are successfully executed, the system will reboot automatically. If you did not remove the diskette at the beginning of the installation, you will be prompted to do it now.

If NetView DM/2 client code is installed on the workstation, enter:

```
CDM STATUS
```

NetView DM/2 is normally started automatically by STARTUP.COM.
If the CDM agent is not active, then issue:

```
CDM START
```

2. At the CC Server perform the following steps:

- Verify that the CC client is attached to the CC server.

Issue the command:

```
CDM LIST_WS
```

If the client workstation is booted from diskettes, status should be
RUNNING.

If NetView DM/2 is already installed, it will have a status of *ACTIVE*.

10.11.4 Install Change Files on Client Workstations

Once connectivity is established between CC server and CC client, the change files can be installed. Prior to installing the change files, the image files and response files should have already been installed on the CC server.

The following steps show you how to install the change files created earlier in this section.

1. Verify that the CC client is attached to the CC server.

(This can be done through the command line by issuing `CDM LIST_WS` or through the PM dialog)

2. Return to the CDM Catalog window.
3. Initiate software installation on the CC client:

At the CDM Catalog window, select the following:

ITSCAUS.OS2WARP4.PREPWS.CMD.REF.1

Click on **Selected** from the action bar to display the pull-down menu. Select **Install** from the pull-down menu. The Install window will be displayed with the selected object listed. Select **Install**. The PREPWS command file will be run on the client and prompt you to remove the diskette. Do not remove the diskette at this stage because the PREPWS.CMD writes to it. It will prompt you to insert Diskette 0 once the partitioning is complete; insert diskette 0 and press **Enter**.

To install the operating system, MPTS and the NVDM/2 agent :

At the CDM Catalog window, select the following:

ITSCAUS.OS2WARP4.INST.REF.4

ITSCAUS.MPTS.INST.REF.5

ITSCAUS.NVDMCLI.INST.REF.2.1

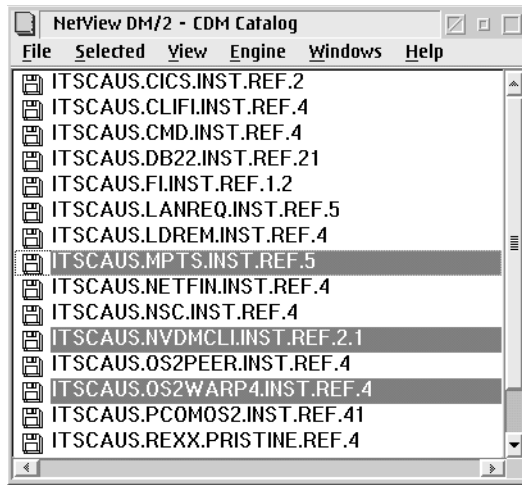


Figure 112. Selecting Software in the CDM Catalog Window

4. Click on **Selected** from the action bar to display the pull-down menu.

Select **Install** from the pull-down menu. The Install window will be displayed with all the selected objects listed.

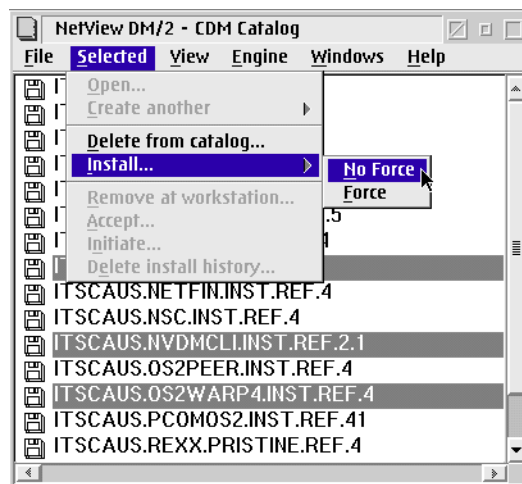


Figure 113. CDM Catalog Install Window

Note: If you are uncertain whether or not the selected software exists on the target workstation, select **Force** as installation method. NetView DM/2

catalogs distributed software in the DB2 database, thus preventing a software product from being distributed more than once. Force overrides this.

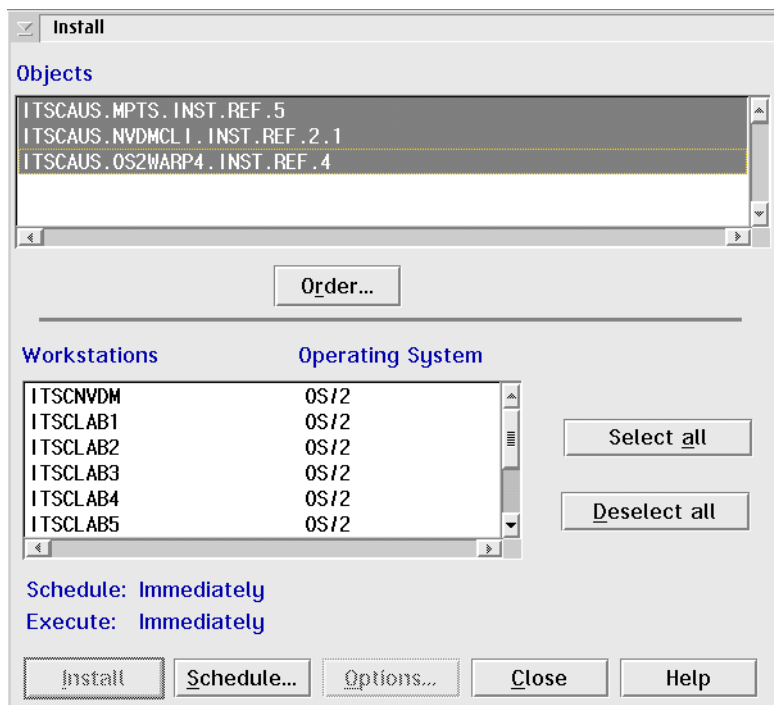


Figure 114. Install Window

5. Define the installation order.

Click on **Order...** to display the Install Order window.

Select **ITSCAUS.OS2WARP4.INST.REF.4**

Select **Up** to move it up to the first position.

Move other objects to arrange in this order:

ITSCAUS.OS2WARP4.INST.REF.4
ITSCAUS.MPTS.INST.REF.5
ITSCAUS.NVDMCLI.INST.REF.2.1

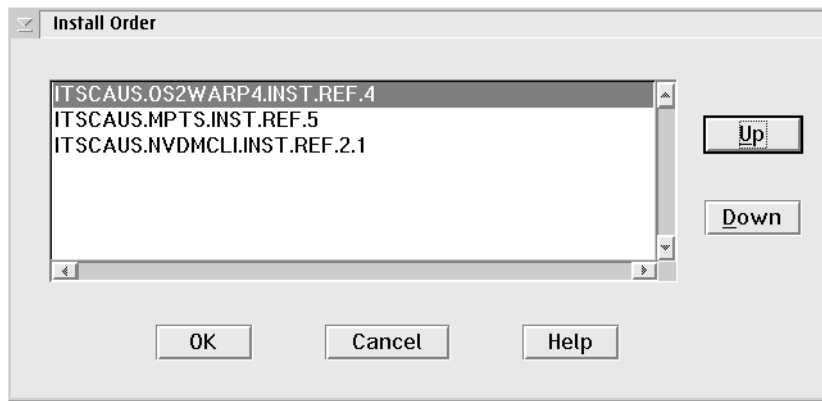


Figure 115. Install Order Window

6. Select **OK** to return to the Install window.

Select client's name from the workstation list to install the objects on client workstation.

7. Define installation options.

Select **Options** from the Install window. The Options screen will be displayed.

Select **Install as a coreq group**.

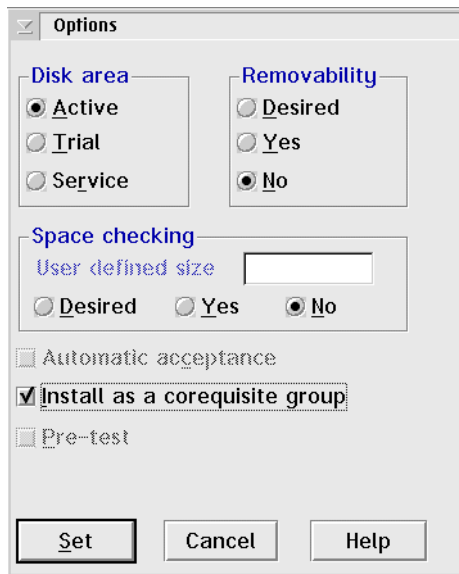


Figure 116. Options Window: Corequisite Group

8. The purpose of "corequisite groups" is to bundle the installation requests of several objects together into one request. Reboot requests of the single objects are queued until a change file with the statement `PhaseEnd=Yes` or the last object of the corequisite group is installed.

Corequisite groups may consist of a maximum of seven change files; they are used to install several pieces of software that depend on each other. If one of the installs in a coreq group fails, the complete group will receive the status "failed", even if installs prior to the failed one were successful.

Click on **Set** to return to the Install window.

9. Select **Install** to execute the command.

You will receive a message popup; check if everything went OK and select **OK** to continue.

Select **Close** at the Install screen to return to the Catalog menu. You may now select other software packages, for example Feature Installer (FISSETUP), OS/2 Warp 4 Installation Part 2 (CLIFI) and FixPak4 (XR_M004) to be distributed to the workstation(s).

10. Check the status of the install request for your client.

At the CDM Catalog window:

Select **Engine** to display the menu.

Select **All Pending Requests** to display the Pending Request menu.

Select client's name.

Select **Details** to display the details.

10.11.5 Using the CDM Install Command

As an alternative to the NetView DM/2 dialog, you can use NetView DM/2 line commands to initiate the install and to check the status.

1. To verify that the CC client is attached to the CC server, issue the following command:

```
CDM LIST_WS
```

The client workstation should be listed as `RUNNING`.

To Install a single change file (meaning No Corequisites), issue:

```
CDM INSTALL ITSCAUS.OS2WARP4.INST.REF.4 /WS:<client name>
```

2. To install change files in a corequisite group, issue the following single-line command:

```
CDM INSTALL ITSCAUS.OS2WARP4.INST.REF.4+  
ITSCAUS.MPTS.INST.REF.5+ITSCAUS.NVDMCLI.INST.REF.2.1 /WS:<client name>
```

Note: This command should be entered as a single command. The plus (+) signs are part of the command and mark it as a coreq group. Do not leave any blanks before or after the plus (+) sign.

3. To check the status of the install request for the client:

```
CDM LIST_REQ /WS:<client name>
```

This `LIST_REQ` command lists all pending requests for the client workstation.

See the online command reference `CDM.INF` located in the `\NVDM2V21` directory or the NVDM/2 manuals for further information on the line commands.

Chapter 11. TME 10 Software Distribution 3.1.3 for OS/2

This chapter is intended to provide information on software distribution of OS/2 Warp 4 using Tivoli's TME 10 SD 3.1.3 for OS/2 originally ported from NetView DM/6000 to OS/2.

The objective of this chapter is to describe the environment and the tasks required to install OS/2 Warp 4 to a pristine client and CID-enabled products remotely, using TME 10 Software Distribution 3.1.3 (SD4OS2).

In order to illustrate the various phases of setup and installation, we will use example scenarios to install OS/2 Warp 4, MPTS and the Software Distribution client over NetBIOS and TCP/IP.

11.1 Environment

The following describes the general environment used for implementing this scenario.

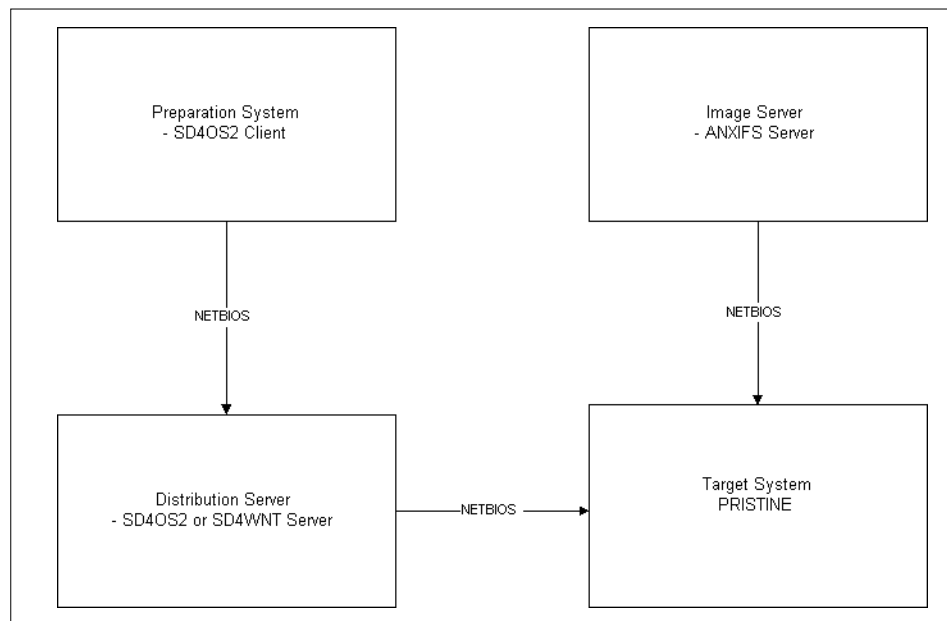


Figure 117. TME 10 SD 3.1.3 NetBIOS Implementation Environment

- **Preparation Server**

This is the system where objects and images are prepared and packaged. It must have the same operating system as the target workstation and it must have an SD4OS2 client installed.

- **Distribution Server**

The Distribution Server has the software distribution server product installed. In this scenario, it could be Software Distribution for NT or for OS/2.

- **Image Server**

The system that provides the redirection support for the agent (pristine agent files, CID code Images, CID response files, CID log files, and command procedures)

Note: For the purposes of these examples, the Image server must be an OS/2 system because the ANXIFS redirection server only runs on OS/2.

- **Target System**

The system where CID-enabled software will be installed.

For this scenario, these are the actual systems that will be used:

- SDSERV1
 - Preparation Workstation
 - Distribution Server
 - Image Server
- CLIENT1
 - Target System

11.2 Installation Steps for OS/2 Distribution Server

The following list shows the installation of the TME 10 SD 3.1.3 software distribution server.

1. Start the installation from the CD-ROM or redirected drive. The installation program, INSTALL.EXE, resides in the SD4OS2\IMAGES directory.

A welcome window of the installation program is displayed as shown in Figure 118.

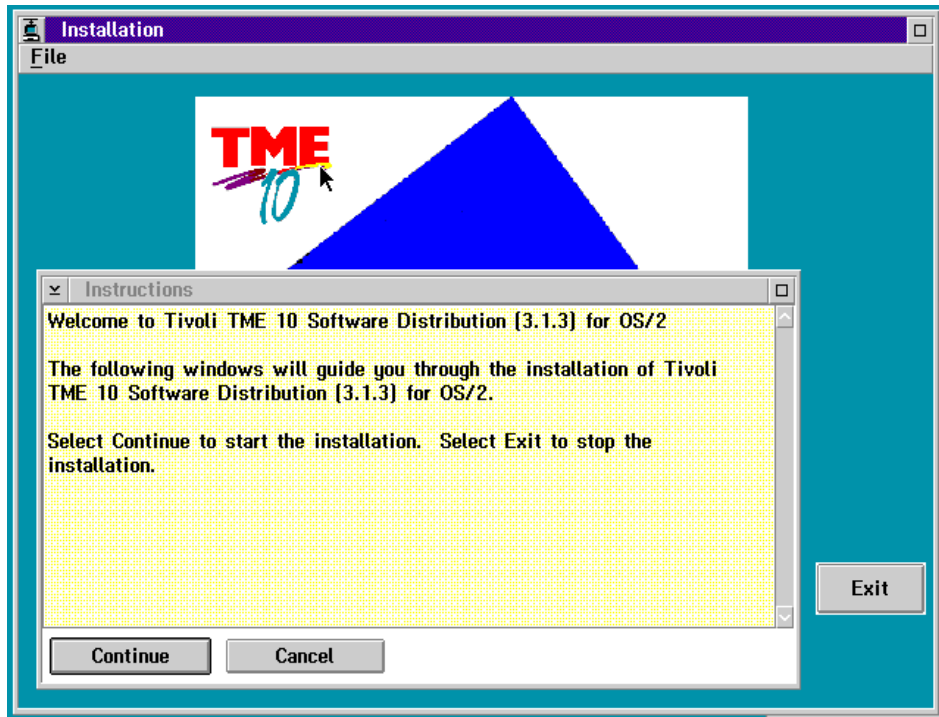


Figure 118. TME 10 Software Distribution 3.1.3 Installation Welcome Screen

2. Select **Continue**.

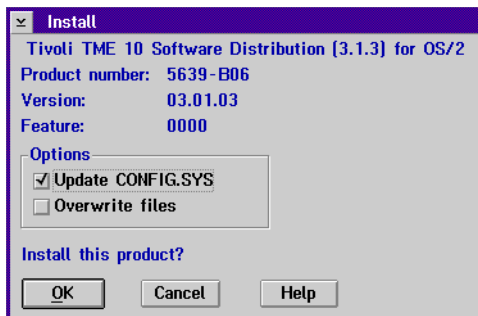


Figure 119. Install Window

3. At the Install window, mark the box that says **Update CONFIG.SYS** and click on **OK** to continue.
4. Select the components you wish to install and the drive on which you want to install the product. We recommend installing the documentation.

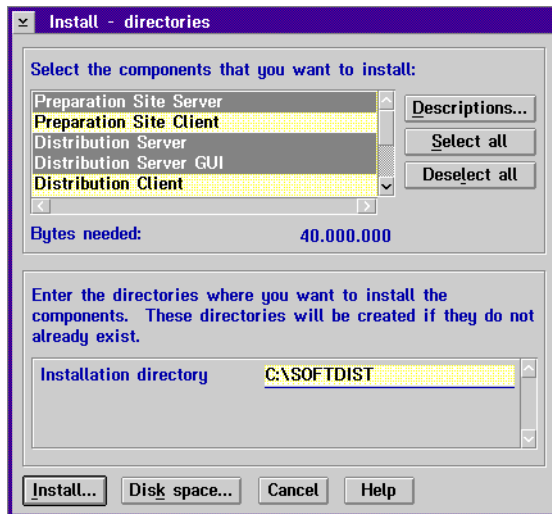


Figure 120. Install - Directories

5. Select **Install**, the Install-progress window will be displayed.

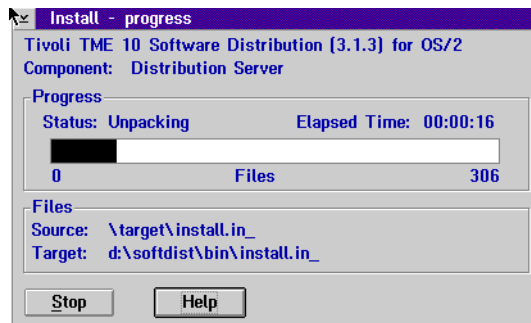


Figure 121. Install Progress Window

6. Use the general page to name the system and choose the communication protocol you want to use. You can use any combination of the communication protocols to connect clients, servers and host systems.

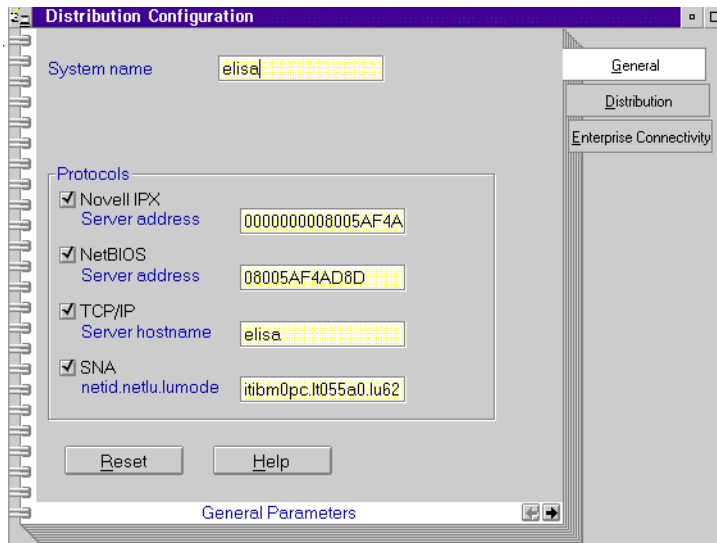


Figure 122. Distribution Configuration - General Window

Important

The name of the system is case-sensitive.

7. On the Distribution page, you can change the directories for the installation:

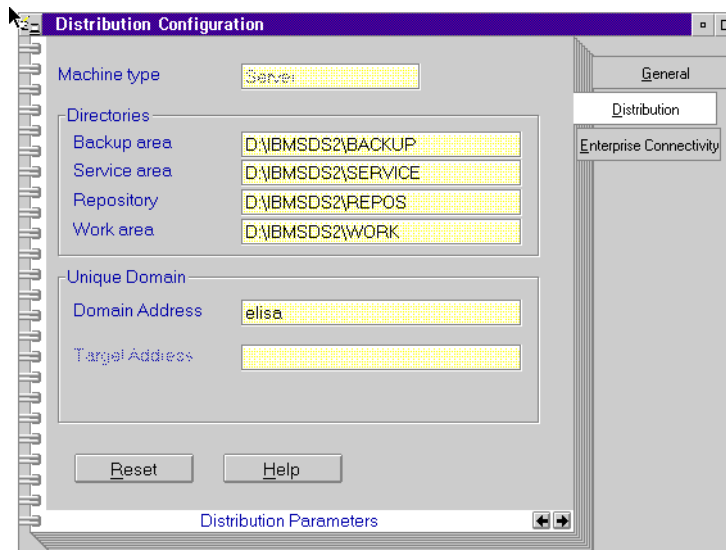


Figure 123. Distribution Window

- | | |
|----------------|--|
| Backup area | The area where previous versions of applications are kept. This entry applies to applications installed through the software distribution on this machine. |
| Service area | The temporary area for installations. |
| Repository | The subdirectory for profiles and software objects. |
| Work area | The temporary area used by software distribution during the resolution of change management requests. |
| Domain Address | If you connect to a distribution server on MVS or AIX, this is the name of the domain to which this software distribution server's clients belong. |

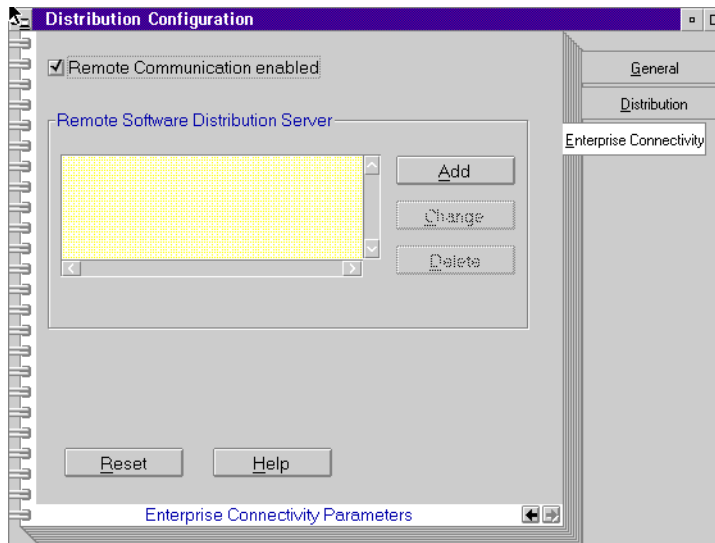


Figure 124. Enterprise Connection Window

8. The Enterprise Connectivity page is used for the connection to NetView Distribution Manager for MVS or AIX distribution server. As we are not using a Host system in our scenario, we uncheck the **Remote Communication enabled** option.
9. Once all the necessary changes are made, close the notebook to save the configuration parameters.

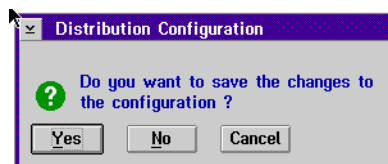


Figure 125. 'Save' Window

A message window will be displayed informing you that the parameters will be saved and that the changes will be applied after the next reboot.

During the TME10 SD installation, a default user ID *root* is defined without a password. This user ID can be used on the server and on the clients to perform administration tasks. It is advisable to change this user ID and/or add a password. See the TME 10 Software Distribution for OS/2 documentation for more details.

Setting Up Password for Root

```
NVDM UPDPWD root -n <pwd/pwd>
```

The password *<pwd>* must be between six and eight characters long.

If you assign a password to *root* you must also add the password to the following files :

- The Start and Stop TME10 Distribution Properties notebook in the TME10 Software Distribution folder
- The FNDUPD.CMD
- The FNDINV.CMD
- The FNDMIG.CMD
- The FNRCH.CMD

These command files can be found in the \SOFTDIST\BIN directory. Add the `-p <password>` to the statement that contains the `NVDM` command.

11.3 Editing the Base Configuration File

The basic configuration of the server is done during the installation process. You will find the base configuration file `NVDM.CFG` in the \SOFTDIST directory.

This file is in text format. Each line starts with one of the keywords described in Table 24 on page 476. A sample configuration file is displayed in “Sample Base Configuration File” on page 480.

The keywords must be entered in uppercase and followed by a colon. Each keyword, except for `PROTOCOL`, can be used only once. The order of the keywords is not important; blanks and comment lines may be included. Comment lines must begin with a number sign (#).

To make changes to the base configuration file, you must stop and restart the server.

Note: If the entries in the base configuration file are incorrect, the product may not start.

11.4 Image/Directory Setup

The following section illustrates setting up images and directories.

11.4.1 Setup Activities

This document assumes that your product images, response files, log files, and samples will be placed in the SDSERV1, following subdirectory structure on the D: drive of SDSERV1. Adjust directory names to your needs if necessary.

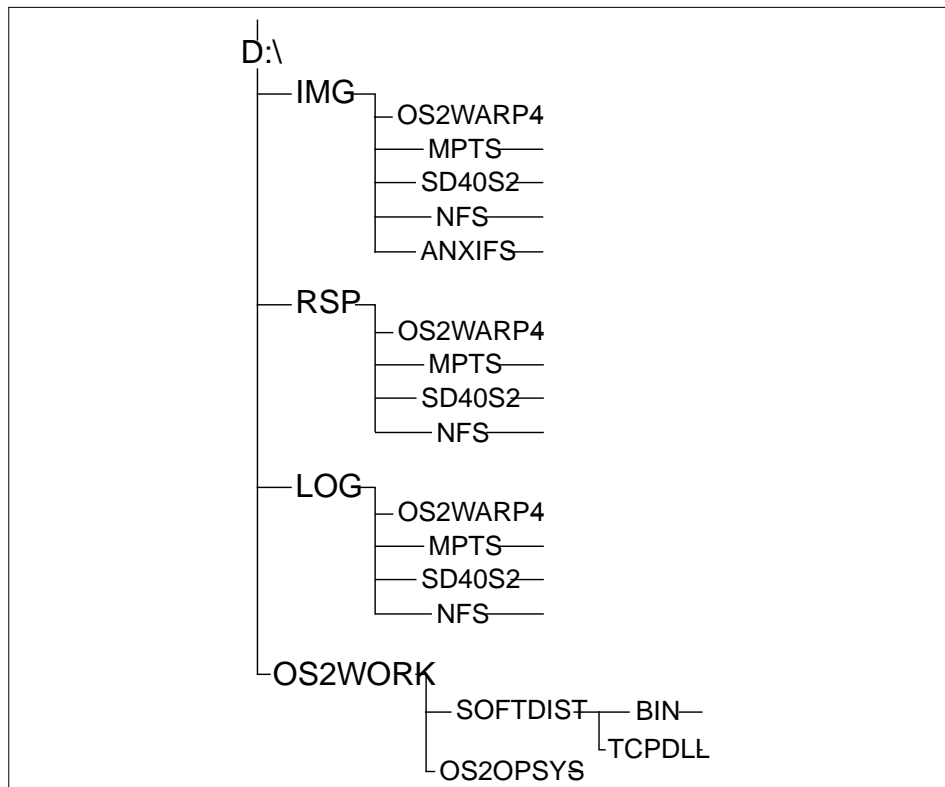


Figure 126. TME 10 SD 3.1.3 Directory Structure

Notes:

1. NFS images, response files, and directories are only needed for the TCP/IP-based scenarios.
2. SD4OS2 NetBIOS redirector images are only needed for the NetBIOS-based scenarios.

3. The copy of an installed SD4OS2 client is only needed for the pristine scenarios.
4. The TCPDLLs (needed to start the SD4OS2 client) are only needed for the pristine TCP/IP scenario.

The following steps describe how to set up the images and directories:

- Create Base Subdirectories

Execute the following commands to create the subdirectories:

```
D:\
MD CID
CD CID
  MD IMG IMG\OSWARP4 IMG\MPTS IMG\SD4OS2 IMG\ANXIFS IMG\NFS
  MD RSP RSP\OS2WARP4 RSP\MPTS RSP\SD4OS2 RSP\NFS
  MD LOG LOG\OS2WARP4 LOG\MPTS LOG\SD4OS2 LOG\NFS
  MD OS2WORK OS2WORK\OS2OPSYS OS2WORK\SOFTDIST
  CD OS2WORK\SOFTDIST
  MD BIN TCPDLL
```

Figure 127. Create Base Subdirectories

- Copy the OS/2 Warp 4 CID utilities to SDSERV1

The following OS/2 Warp 4 utilities are required to perform this installation and must be copied to the \CID\IMG\OS2WARP4 directory if this has not been done previously.

- SEIMAGE.EXE
- SEDISK.EXE
- SEMAINT.EXE
- SEINST.EXE -
- RSPINST.EXE

Place the OS/2 Warp 4 CD-ROM in the CD-ROM drive and execute the following commands:

```
UNPACK E:\OS2IMAGE\DISK_7\CID D:\CID\IMG\OS2WARP4
UNPACK E:\OS2IMAGE\DISK_7\REQUIRED D:\CID\IMG\OS2WARP4 /N:RSPINST.EXE
```

(where E: is the CD_ROM drive)

- Copy the OS/2 Warp 4 product images to SDSERV1

If you are using a CD-ROM, place the CD in the CD-ROM drive and enter the following command:

```
XCOPY E:\OS2IMAGE\*. * D:\CID\IMG\OS2WARP4 /S /E
```

This will place all the required OS/2 Warp 4 product diskettes in the appropriate directories.

Note: In order to save disk space, the SYM_x subdirectories can be deleted.

- Copy the MPTS images to SDSERV1 (Pristine Scenarios, only)

To create the MPTS image from the OS/2 Warp 4 CD-ROM, enter the following command:

```
XCOPY F:\CID\IMG\MPTS\*. * D:\CID\IMG\MPTS /S /E
```

- Copy the NFS images to SDSERV1 (TCP/IP Scenarios, only)

Place the diskette with NFS kit for TCP/IP V2.0 into the A: drive and enter:

```
D:  
CD \CID\IMG\NFS  
XCOPY A:*. * /S /E
```

Place the diskette with NFS CSD UN57064 V2.0 into the A: drive and enter:

```
D:  
CD \CID\IMG\NFS  
XCOPY A:*. * /S /E
```

Obtaining NFS CSD

The latest NFS Corrective Service Diskette (CSD) is available on the following FTP server accessible through the Internet:

```
service.boulder.ibm.com
```

Logon with the user ID of `anonymous` and type your email ID as your password. Change to the `/ps/products/tcpip/fixes/v2.0os2/un57064` directory. Get the `u57064d1.dsk` file in binary format and use the `LOADDSKF` diskette unpacker to unpack this file onto diskette.

- Copy the ANXIFS images to SDSERV1 (NetBIOS scenarios, only)
To copy the ANXIFS images from the CD-ROM for TME10 Software Distribution V3.1.3, enter the following command:

```
PKUNZIP2 -d F:\SD4OS2\PRISTINE\ANXIFS.ZIP D:\CID\IMG\ANXIFS
```

If you received the ANXIFS.ZIP file separate from the product, place the diskette with this file into your diskette drive and enter the following commands:

```
D:  
CD \CID\IMG\ANXIFS  
PKUNZIP2 -D A:\ANXIFS.ZIP
```

Note: Ensure that your version of ANXIFS.ZIP has a file called `CLIENT\ANXREQ.EXE` (instead of `CLIENT\APICLT.EXE`). Pre-release versions of ANXIFS.ZIP had slightly different file names, and they will not work with these scenarios.

The ZIP contents will be unpacked into `D:\CID\IMG\ANXIFS`

- Copy OS2OPSYS.ZIP to SDSERV1

Place the diskette with this samples file into your diskette drive and enter the following commands:

```
D:  
CD \CID\OS2WORK\OS2OPSYS  
PKUNZIP2 OS2OPSYS.ZIP
```

This file is shipped as part of these scenarios.

- Update the ANXIFS product images (NetBIOS scenarios, only)
Copy update files from the SAMPLES directory to the ANXIFS directory:

```
D:  
CD \CID\IMG\ANXIFS\CLIENT  
ERASE INSTALL.CMD  
ERASE FNDVNB.CMD  
COPY D:\CID\OS2WORK\OS2OPSYS\ANXIFS.CMD
```

INSTALL.CMD and FNDVNB.CMD are erased to prevent their accidental usage. They are older versions that are not used in these scenarios.

ANXIFS.CMD is a newer version of the ANXIFS client install routine that is used as part of the pristine install.

- Copy the SD4OS2 files to the target directory

SD4OS2 is only shipped on CD-ROM; so this section must be done from a system with a CD-ROM drive.

Insert the TME 10 Software Distribution CD-ROM (either the September, 1996 GA version, or the October 1996 Refresh version) and enter:

```
D:  
XCOPY E:\SD4OS2\IMAGES\*. * D:\CID\IMG\SD4OS2 /S /E
```

where **E:** is the drive letter of the CD-ROM drive

- Update SD4OS2 images with latest CSD Read the appropriate READ.ME for information on installing the latest available fix. For these scenarios, XR21382 was applied to the TME 10 SD 3.1.3 images by running SD4OS2.EXE.
- Copy the BIN directory from an installed SD4OS2 Client onto SDSERV1 (Pristine Scenarios, only).

You must install the SD4OS2 client code (at the same level used in these scenarios) onto a client system, and then `xcopy` the \SOFTDIST\BIN directory into D:\CID\OS2WORK\SOFTDIST\BIN.

These client files are not SD4OS2 images. They are SD4OS2 client files from an installed SD4OS2 client, which are then copied to a directory on the server (D:\CID\OS2WORK).

11.4.2 Additional Hints and Tips

The following hints and tips are helpful for software distribution scenarios with TME 10 SD 3.1.3:

1. Ensure that you copy the BIN directory from an installed SD4OS2 client..
2. Do NOT copy the BIN directory from an installed SD4OS2 server. The server executables will not work in the pristine scenario.
3. Ensure that your D:\CID\OS2WORK directory contains the file DPFUPD.EXE. If it does not, copy this file from the \SOFTDIST\BIN directory on your SD4OS2 server (SDSERV1).
4. Ensure that your SD4OS2 images are at the same level of SD4OS2 that are in your \CID\IMG\SD4OS2.
 1. If your \CID\IMG\SD4OS2 images are at the GA level (meaning dated August 1996, version 313), then your \CID\OS2WORK\SOFTDIST\BIN client executables must also be at the GA level.
 2. If your \CID\IMG\SD4OS2 images are at the CSD level (meaning dated October 1996, version 3131), then your \CID\OS2WORK\SOFTDIST\BIN client executables must also be at the CSD level.

11.5 Setup Activities

The following section illustrates setup activities at the server such as preparing response file, server and client initialization files, and creating boot diskettes.

11.5.1 Prepare Response Files at SDSERV1

Basically, product response files used by TME 10 SD 3.1.3 are the very same used by NetView DM/2 or LAN CID Utility. See Chapter 7, "Creating Response Files" on page 181, for a detailed discussion on response files.

For pristine installations scenarios make sure the following requirements are met:

- OS/2 Response File OS2.RSP

The following keywords must be set to the following values:

```
ExitOnError=1
FormatPartition=1
MigrateConfigfiles=0
RebootRequired=1
```


Note: If `FormatPartition` is set to 0, the installation program tries a migration.

- MPTS Response File MPTSNB.RSP

This file is in the `D:\CID\OS2WORK\OS2OPSYS` subdirectory. It is not copied into the `\CID\RSP\MPTS` directory because it is built into the change file.

- TME10 SD 3.1.3 Client Response File SD4OS2NB.RSP

This file is in the `D:\CID\OS2WORK\OS2OPSYS` subdirectory. It is not copied into the `\CID\RSP\SD4OS2` directory because it is built into the change file (to allow the use of a single response file with replaceable tokens).

- TME 10 SD 3.1.3 Client Modification File

Issue the following commands:

```
D:  
CD \CID\RSP\SD4OS2  
COPY \CID\OS2WORK\OS2OPSYS\STARTUP.MOD
```

This file removes the pristine client start command from `STARTUP.CMD` (after the installation of the `SD4OS2` client).

11.5.2 Configure Redirection Support

The NetBIOS redirection support is provided by `ANXIFS`, which is provided as part of `SD4OS2` server. `ANXSRV.INI` must be customized before starting the server redirection support, and `ANXCLT.INI` must be customized before creating client boot diskettes.

1. Modify `ANXSRV.INI`

Edit `D:\CID\IMG\ANXIFS\SERVER\ANXSRV.INI` and update it as follows:

```

Adapter=0;
Numclients=20
SERVERNAME=SDSERV1
permitwrite=yes

;REMAGT is the alias for the SD Client code that the pristine
;target needs access to in order to complete the install
;C:\cid\pristine is ;where this is located

ALIAS=RW, SINGLE, REMAGT, D:\CID\OS2WORK\SOFTDIST

;CODESRV is the alias for the CID directory, D:\CID is the directory
;where the images, the response files and log files are located

ALIAS=RW, SINGLE, CODESRV, D:\CID

```

Figure 128. TME 10 SD 3.1.3 ANXSRV.INI File

The highlighted lines indicate changes from the original version.

This file should be customized to reflect the directory structure in your environment and the aliases you wish to use.

Note: This file is used on SDSERV1 to define the redirection server configuration.

2. Modify ANXCLT.INI

Edit D:\CID\IMG\ANXIFS\CLIENT\ANXCLT.INI and update it as follows:

```

;Adapter Number supported 0-1

adapter=0

;Max numbers of attach

numattach=4

;Clientname not mandatory if this key is not set a random name will
;be used
;clientname=ff
;Alias to be attached
;z is the disk letter
;ANXS01 is the Anxifs Server Name
;REMAGT is the alias for the remote Agent
;CODESRV is the alias for the cid directory where the images, response
;files and log files are loaded

ATTACH=Z,SDSERV1,REMAGT
ATTACH=X,SDSERV1,CODESRV

```

Figure 129. TME 10 SD 3.1.3 ANXCLI.INI File

The highlighted lines indicate changes from the original version.

This file should be customized to reflect the directory structure in your environment, the aliases you wish to use, and the SD4OS2 or SD4WNT server name.

Note: This file is used on CLIENT1 to define the redirection client configuration.

3. Start the Redirection Server

The ANXIFS redirection server can be started by issuing the following commands:

```

D:
CD D:\CID\IMG\ANXIFS\SERVER
APISERV ANXSRV.INI

```

11.5.3 Additional Sample Profiles

The following profiles are samples only. For more information on the utilities, such as Feature Installer (CLIFI), Remote Printer Install (RMPI), and Desktop Shuffler (CLNDESK), and other OS/2 Warp 4 subcomponents, such as TCP/IP, OS/2 Peer Services, and so on, refer to the following chapters:

- Chapter 4, “Utilities to Ease Remote Installation” on page 75
- 10.9, “Creating Change Files” on page 361

The logic of NetView DM/2 change profiles can be easily adapted to TME 10 SD 3.1.3 change profiles.

For more information on creating profiles, refer to the TME 10 SD 3.1.3 documentation.

```

GLOBAL NAME:          IBM.WARP4.XR_M004.REF.4
DESCRIPTION:          " Installation for OS/2 Warp 4 FixPak 4"
CHANGE FILE TYPE:    OS2CID
INSTALL PROGRAM:

PROGRAM NAME:        X\IMG\CSD\KICKER\FSERVICE.EXE
PARAMETERS:          /S:X:\IMG\CSD\XR_M004 /R:$(RSPFILE)
                    /L1:X:\LOG\XR_M004.LOG /CID
RESPONSE FILE:      E:\RSP\XR_M004\RESPONSE.FIL

```

Figure 130. TME 10 SD 3.1.3 XR_M004.PRO Change Profile

Note: The PARAMETERS line must be a single line.

```

GLOBAL NAME:          IBM.CLIFI.INSTALL.REF.1
DESCRIPTION:          IBM CLIFI INSTALL
CHANGE FILE TYPE:    OS2CID
INSTALL PROGRAM:

PROGRAM NAME:        X:\IMG\OS2WARP4\CLIFI.EXE
PARAMETERS:          /A:C /F:C:\OS2\INSTALL /S:X:\IMG\OSWARP4
                    /R:C:\OS2\INSTALL\FIBASE.RSP /R2:$(RSPFILE)
                    /L1:X:\LOG\OS2WARP4\$(TARGET).FI1
                    /L2:X:\LOG\OS2WARP4\$(TARGET).FI2
RESPONSE FILE:      X:\RSP\OS2WARP4\$(TARGET).RSP

```

Figure 131. TME10 SD 3.1.3 CLIFI.PRO Change Profile

Note: The PARAMETERS line must be a single line.

```

CHANGE FILE TYPE:      OS2CID
DEFAULT TOKEN:         TargetDir=C:
INSTALL PROGRAM:

PROGRAM NAME:  X:\IMG\RMPI\RINSTPRN.EXE
PARAMETERS:   /S:X:\IMG\RMPI
              /DSC:X:\IMG\OS2WARP4\PMDD_1\PRDESC.LST
              /DRV:X:\IMG\OS2WARP4\PMDD_1\PRDRV.LST
              /WPR:C:\OS2\MDOS\WINOS2\SYSTEM\DRVMAP.INF
              /R:X:\RSP\RMPI\$(TARGET).RSP
              /L1:X:\LOG\RMPI\$(TARGET).L1

```

Figure 132. TME10 SD 3.1.3 RMPI.PRO Change Profile

Note: The PARAMETERS line must be a single line.

```

GLOBAL NAME:          IBM.ICON.SHUFFLER.REF.1
DESCRIPTION:          "Sample Icon Shuffler Profile (CLNDESK.EXE)"
CHANGE FILE TYPE:    OS2CID
INSTALL PROGRAM:

PROGRAM NAME:  X:\EXE\ISHUFFL.COMD

```

Figure 133. TME10 SD 3.1.3 SHUFFLER.PRO Change Profile

11.5.4 Prepare Pristine Client Boot Diskettes

Three boot diskettes are required for the OS/2 Warp 4 NetBIOS pristine installation scenario:

- Diskette 0
 - Diskette 1
 - Diskette 2
1. Create the initial OS/2 boot diskettes

Insert Diskette 0 into Drive A: and issue the following command:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A:
```

When prompted, insert Diskette 1 and then Diskette 2.

PCMCIA Support

If you are are creating boot diskettes for ThinkPads, add the `/P:` parameter to the `SEDISK` command and pass the line number in `PCMCIA.TBL` that describes the ThinkPad you want boot diskettes for. Note that `PCMCIA.TBL` is OS/2-version dependent and can be found in `\OS2\INSTALL` or in `\DISK_3\BUNDLE` in case of OS/2 Warp 4.

For example, to install OS/2 Warp 4 on a ThinkPad 760ED (Line 36 in `PCMCIA.TBL` from OS/2 Warp 4 represents this ThinkPad), issue:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A: /P:36
```

2. Add standard NetBIOS support to the last boot diskette.

Leave the last diskette in Drive A: and issue:

```
D:\CID\IMG\MPTS\THINLAPS D:\CID\IMG\MPTS A: IBMTOK.NIF
```

When installing OS/2 Warp 4, you will be prompted to insert Diskette 1 in order for `THINLAPS` to update the `CONFIG.SYS`.

Consider Correct Ring Speed

Specify the correct NIF file for your adapter. For example, use `IBMTOKCS.NIF` for PCMCIA token-ring adapters.

You may need to edit the `PROTOCOL.INI` in these cases. For example, for `IBMTOKCS.NIF`, you may need to change the `RINGSPEED` keyword if the standard ring speed is not 16 MBit/sec.

3. Install the TME 10 Sd 3.1.3 (SD4OS2) client code on the last diskette.

Leave the last diskette in Drive A: and issue the following single-line commands:

```
D:
CD D:\CID\OS2WORK\OS2OPSYS
MKOS2NB D:\CID\OS2WORK\OS2OPSYS D:\CID\IMG\OS2WARP4
D:\CID\IMG\ANXIFS\CLIENT
```

where:

`D:\CID\OS2WORK\OS2OPSYS` The directory for the pristine samples.

`D:\CID\IMG\OS2WARP4` The directory for OS/2 images.

`D:\CID\IMG\ANXIFS\CLIENT` The directory for the ANXIFS client images.

Note: The `MKOS2NB` line must be a single line.

For OS/2 Warp 4, switch to Diskette 1 when prompted.

This batch file will:

- Erase unneeded files from the boot diskette
UNPACK.EXE, UNPACK2.EXE
- Copy files from the ANXIFS client images:
OSO001.MSG, ANXREQ.EXE, ANXIFPID.SYS, ANXIFCOM.SYS,
ANXIFPID.SYS, ANXIFCOM.SYS, ANXIFCOM.IFS, ANXCLT.INI
- Copy files from the PRISTINE samples:
FDISK.DAT, PREPDSK.CMD, FORMATC.DAT, FORMDSK.CMD,
STARTNB.CMD, FNDVNB.CMD
- Copy VDISK.SYS from the OS/2 images
- Modify lines in CONFIG.SYS using DPFUPD:
Change the OS/2 shell line to read:
SET OS2_SHELL=CMD.EXE /K FNDVNB.CMD
Add .: (period and semicolon) to the beginning of LIBPATH, PATH, and
DPATH.
Add lines in CONFIG.SYS:
DEVICE=\ANXIFPID.SYS
DEVICE=\ANXIFCOM.SYS
DEVICE=\VDISK.SYS 700, ,
IFS=ANXIFCOM.IFS

Target-specific Parameters

The NetBIOS pristine boot diskettes do not contain target-specific configuration information and can be used concurrently on multiple targets.

(The NetBIOS protocol does not contain target-specific parameters, and the ANXIFS redirector creates a unique computer name. Thus, the single set of boot diskettes can be used concurrently).

11.6 Create the Change Files

The profiles for each object that will be installed at the target have to be created at the software distribution server. You can find sample profiles in D:\IMG\OS2WORK\OS2OPSYS.

```

GLOBAL NAME: IBM.OS2WARP.REF.4000
DESCRIPTION: IBM OS/2 4.0 Pristine
CHANGE FILE TYPE: OS2CID
DEFAULT TOKEN: TargetDir=C:
INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\OS2WARP4\SEINST.EXE
PARAMETERS: /S:X:\IMG\OS2WARP4 /T:A:\ /B:${TargetDir}
/R:X:\RSP\OS2WARP4\OS2.RSP /L1:X:\LOG\OS2WARP4\$(TARGET).L1

```

Figure 134. OS/2 Warp 4 Pristine Installation Profile WARP4.PRO

Note: The PARAMETERS line must be a single line.

```

GLOBAL NAME: IBM.MPTS.NETBIOS.REF.8400
DESCRIPTION: IBM MPTS
CHANGE FILE TYPE: OS2CID
INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\MPTS\MPTS.EXE
PARAMETERS: /E:MAINT /S:X:\IMG\MPTS /T:${TargetDir}\ /TU:${TargetDir}\
/R:${RSPFILE} /L1:X:\LOG\MPTS\$(TARGET).L1
RESPONSE FILE: X:\OS2WORK\OS2OPSYS\MPTS.NB.RSP

```

Figure 135. MPTS Pristine Install Profile MPTS.NB.PRO (With NETBIOS)

Note: The PARAMETERS line must be a single line.

```

GLOBAL NAME: IBM.ANXIFS.REF.1
DESCRIPTION: IBM anxifs client
DEFAULT TOKEN: TargetDir=C:
CHANGE FILE TYPE: OS2CID
INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\ANXIFS\CLIENT\ANXIFS.COM
PARAMETERS: X: ${TargetDir} Z:

```

Figure 136. ANXIFS Pristine Install Profile ANXIFS.PRO


```

GLOBAL NAME: IBM.SD4OS2.CLINB.REF.1
DESCRIPTION: SD4OS2 Client -- Over NetBIOS
CHANGE FILE TYPE: OS2CID
POST-INSTALL: DPFUPD.EXE /MOD:X:\RSP\SD4OS2\STARTUP.MOD
/DRIVE:$(BOOTDRIVE)
INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\SD4OS2\INSTALL.EXE
PARAMETERS: /S:X:\IMG\SD4OS2 /R:$(RSPFILE)
/L1:X:\LOG\SD4OS2\$(TARGET).L1
/L2:X:\LOG\SD4OS2\$(TARGET).L2 /X
RESPONSE FILE: X:\OS2WORK\OS2OPSYS\SD4OS2NB.RSP

```

Figure 137. SD4OS2 Client Install Profile SD4OS2NB.PRO

Note: The `PARAMETERS` line must be a single line.

To create the objects, issue the following commands:

```

D:
CD \CID\OS2WORK\OS2OPSYS
NVDM BLD WARP.PRO
NVDM BLD MPTSNB.PRO
NVDM BLD SD4OS2NB.PRO
NVDM BLD ANXIFS.PRO

```

The objects are created, and the global names are added to the catalog.

11.7 Install the Pristine Client

The following illustrates steps to do to enable software distribution to a pristine client.

11.7.1 Define Client at SDSERV1

Issue the following command to define the pristine client:

```
NVDM ADDTG CLIENT1 -s CLIENT1 -a CLIENT1 -y OS/2 -tp nbi:CLIENT1
```

11.7.2 Define Client Specific Installation Parameters

If you wish to install the pristine client on a drive other than C:, also define a value for `TargetDir`:

```
NVDM ADDPM CLIENT1 -i TargetDir=D:
```

Notes:

1. CLIENT1 by default will install on the C: drive. You only need to specify this override token if you want to install it on another drive letter.
2. Adding this token does not set up the Boot Manager settings that are needed to enable the boot of an alternate partition.

11.7.3 Submit Installation Commands at SDSERV1

Issue the following single-line commands to submit the installation:

```
NVDM INST IBM.OS2WARP.REF.1 IBM.MPTS.NETBIOS.REF.1 IBM.ANXIFS.REF.1 -n -w  
CLIENT1  
NVDM INST IBM.SD4OS2.CLINB.REF.1 -n -w CLIENT1
```

Note: The first two lines form one single command. The `-n` flag indicates that this is a non-removable install. None of these OS2CID objects have BACKUP, REMOVE, or ACCEPT programs defined; so the installs must be non-removable.

11.7.4 Boot CLIENT1 from Diskettes

CLIENT1 can be booted from diskette before or after submitting the installation commands:

1. Insert Diskette 0 in drive A:.
2. Reboot the system.
3. When prompted insert Diskette 1 and press Enter.
4. When prompted insert Diskette 2 and press Enter.
5. (Optional) Partition the hard disk.

Note

The hard disk must be partitioned, and the OS/2 installation drive must be formatted before starting the next step;

These steps could have been done previously (because of a reinstall versus a printine install), could be done manually, or could be done through the command file.

6. (Optional) Format the hard disk.
7. From the OS/2 command prompt, issue the following command:

```
STARTNB CLIENT1 CLIENT1 SDSERV1 SDSERV1 C
```

where:

```
CLIENT1 The target name.
```

CLIENT1 Tthe target address.
SDSERV1 Tthe server name.
SDSERV1 The server address.
C: The target drive for the pristine installation.

Remember

Remember, SD4OS2 is case-sensitive; use the same case in all your definitions.

11.7.5 Monitor Installation Results

The installation commands can be queued at the server before or after starting the pristine agent (through STARTNB.CMD). In either case, the pristine client will do the following:

1. Execute STARTNB.CMD

The install command performs various tasks:

- Establishes where to locate the client code
- Sets up a target directory for the install
- Assigns the variables input at the command line
- Creates a NVDM.CFG file
- Starts the pristine SD4OS2 client

2. When the install command has completed successfully, a screen similar to that below will be shown.

```
**Chosen Values**
(nvdm-client-name):
(nvdm-client-address max 8 chars):
(nvdm-server-name):
(nvdm-server-address max 12 char):
(target-drive-letter 1 char):
Building Service Directory...
Writing Software Distribution Configuration File...
Writing Software Distribution Final Configuration File...
Starting Software Distribution...
Connected to server <server_name>
Task fndcmps has pid<xx>
N/A
```

3. The OS/2 installation will begin, displaying the standard blue installation screens.
4. The boot diskette can be removed after the *Loading System Files. Please wait....* blue installation screen gives way to the Copying Diskette x series of panels.
5. After the OS/2 install completes, the MPTS install will take place. MPTS writes its messages to standard output and SD4OS2 redirects all the standard output to REQUEST.OUT; so you will not see any indication that the MPTS install is taking place.
6. After MPTS is installed, ANXIFS.CMD is run (see Figure) to install the ANXIFS client. The ANXIFS client is installed.

This will copy the ANXIFS client code onto the hard disk, add it to CONFIG.SYS and STARTUP.CMD, and create the FNDPRE.CMD and FNDPOST.CMD batch files that will be used in future scenarios to establish and de-establish redirection.

7. The second NVDM INST command will now begin. This command installs the SDS code on the pristine client and also removes the execution of FNDPRS from the STARTUP.CMD (because it is no longer needed after the real agent is installed). This is done by running DPFUPD.EXE as a post-req along with the following modification file:

```
{$(TargetDir)\STARTUP.CMD}
DeleteLine()
CALL $(TargetDir)\SOFTDIST\FNDPRS.CMD
```

Figure 138. (STARTUP.MOD) - SD4OS2 Installation Modification File

8. The install of SD4OS2 adds the 'Start TME 10 Software Distribution Client' Icon to the OS/2 Startup Folder folder so that it is started automatically on all subsequent reboots.
9. The agent reboots, and the pristine installation is now complete.

11.7.6 Verify Request Completion

At the server SDSERV1, issue the following command:

```
NVDM LSRQ -w CLIENT1
```

to display the status of all requests for CLIENT1. The requests are completed successfully only if the Status shows as successful:

```
Request ID: SDSERV1 root 425 0
Domain: SDSERV1
Target: CLIENT1
Submission time: 10/02/96 05:12 PM
Request type: Install
Object: IBM.OS2WARP.REF.1
Status: Successful
Error severity:0
Error code: 0000:0000
Completion time: 10/02/96 05:40 PM
```

```
Request ID: SDSERV1 root 426 0
Domain: SDSERV1
Target: CLIENT1
Submission time: 10/02/96 05:13 PM
Request type: Install
Object: IBM.SD4OS2.CLINB.REF.1
Status: Successful
Error severity:0
Error code: 0000:0000
Completion time: 10/02/96 05:50 PM
```

Note that for a corequisite installation, only the first global name is shown in the Object field.

A completed request can be deleted by issuing the following command:

```
NVDM ERASEREQ xxx
```

where xxx is the numeric request number. The request numbers from the above examples are 425 and 426.

11.8 OS/2 Pristine Installation over TCP/IP

This scenario describes how to install OS/2 Warp 4 software to a pristine client over TCP/IP using the TME 10 Software Distribution server. The pristine client will be booted through boot diskettes. The following software will be installed on the remote client CLIENT1:

- OS/2 Warp 4
- MPTS configured for TCP/IP
- TME 10 Software Distribution Client for OS/2
- NFS Kit for TCP/IP 2.0

11.8.1 Prepare Response Files at SDSERV1

As mentioned in the previous software distribution scenario, product response files used by TME 10 SD 3.1.3 are the very same used by NetView DM/2 or LAN CID Utility. See Chapter 7, "Creating Response Files" on page 181, for a detailed discussion on response files.

For pristine installations scenarios, make sure the following requirements are met:

- OS/2 Response File OS2.RSP

The following keywords must be set to the following values:

```
ExitOnError=1
FormatPartition=1
MigrateConfigfiles=0
RebootRequired=1
```

Note: If `FormatPartition` is set to 0, the installation program tries a migration.

- TCP/IP-NFS Response File NFS.RSP

Issue the following commands:

```
D:
CD \CID\RSP\NFS
COPY \CID\OS2WORK\OS2OPSYS\NFS.RSP
```

For the NFS install, this one generic response file, NFS.RSP, can be used to install all targets.

```

// Default response file for PRODUCT DISK install

INSTALL_NAME = NFS 1.10 1 1 "Network File System Kit" NFS Kit
INSTALL_NAME = NFSCID 0.25 1 1 "Network File System Kit" NFS TCPIP CID
Install
//INSTALL_NAME = ILR 1.32 1 1 "IBM Library Reader" IBM Library Reader

// Default response file for PRODUCT DISK install

INSTALL_NAME = NFSC 0.99 1 1 "CSD UN57064, NFS Kit" CSD UN57064
INSTALL_NAME = NFSCIDC 0.23 1 1 "CSD UN57064, NFS Kit" CSD UN57064

EXEC = NFS call nfsxt BOOT_DRIVE TARGET_PATH
CONFIGURE = Y
CONFIGSYS = Y
INSTALL_LAPS = N
ATTENDED=N

```

Figure 139. NFS Response File NFS.RSP

- SD4OS2 TCP/IP Client Response File SD4OS2IP.RSP

This file is in the D:\CID\OS2WORK\OS2OPSYS subdirectory. It is not copied into the \CID\RSP\SD4OS2 subdirectory because it is built into the change file (to allow the use of a single response file with replaceable tokens).

- SD4OS2 Client Modification File MPTSTUP.MOD

Issue the following commands:

```

D:
CD \CID\RSP\SD4OS2
COPY \CID\OS2WORK\OS2OPSYS\STARTUP.MOD

```

This file removes the pristine client start command from STARTUP.CMD (after the installation of the SD4OS2 client).

- MPTS TCP/IP Response File MPTSIP.RSP

This file is in the D:\CID\OS2WORK\OS2OPSYS subdirectory. It is not copied into the RSP subdirectories because it is built into the change file (to allow the use of a single response file with replaceable tokens).

11.8.2 Configure Redirection Support

The redirection support in this scenario is provided by an OS/2 NFS server on SDSERV1. However, this NFS server could be on any NFS server capable operating system such as OS/2, AIX, Windows NT, and so forth.

1. Define EXPORTS at SDSERV1

Edit (or create) the C:\MPTN\ETC\EXPORTS file and add the following lines:

```
D:\CID
D:\CID\OS2WORK\SOFTDIST
```

Note: This file is used on SDSERV1 to define the redirection server configuration.

2. Modify Pristine Client Initialization Files

Edit D:\CID\OS2WORK\OS2OPSYS\SDPRISIP.COM and change `xxxxxxx` in the following lines to match the TCP/IP hostname of SDSERV1:

```
@ECHO MOUNT -u0 -g0 z: <xxxxxxx>D:\CID\OS2WORK\SOFTDIST
                                     >> %2\SOFTDIST\FNDPRE.COM
@ECHO MOUNT -u0 -g0 x: xxxxxxxx:D:\CID >> %2\SOFTDIST\FNDPRE.COM
```

- Edit D:\CID\OS2WORK\OS2OPSYS\FNDVIP.COM and D:\CID\OS2WORK\OS2OPSYS\FNDVIPW.COM and change: `xxxxxxx` in the following lines to match the TCP/IP hostname of SDSERV1:

```
MOUNT -u0 -g0 z: xxxxxxxx:D:\CID\OS2OPSYS\SOFTDIST
MOUNT -u0 -g0 x: xxxxxxxx:D:\CID
```

Note: These files are used on CLIENT1 to define the redirection client configuration

PCNFSD

If your NFS server uses the PCNFSD, you need to alter the MOUNT commands to use a user ID and password:

```
MOUNT -l<user> -p<password> x: <server hostname>:D:\CID
```

3. Start (or restart) the NFS Server.

- (If not already started) start the PORTMAP daemon:

```
START PORTMAP
```

- If the NFS server is already running, terminate it.

- Start the NFSD daemon:

```
NFSD
```

11.8.3 Prepare Pristine Client Boot Diskettes (OS/2 Warp 4)

Three boot diskettes are required for the OS/2 Warp 4 - TCP/IP pristine installation scenario:

- Diskette 0
- Diskette 1
- Diskette 2

1. Create the initial OS/2 Boot Diskettes.

Insert Diskette 0 into Drive A: and issue the following command:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A:
```

When prompted, insert Diskette 1 and then Diskette 2.

PCMCIA Support

If you are are creating boot diskettes for ThinkPads, add the /P: parameter to the SEDISK command and pass the line number in PCMCIA.TBL that describes the ThinkPad you want boot diskettes for. Note that PCMCIA.TBL is OS/2-version dependent and can be found in \OS2\INSTALL or in \DISK_3\BUNDLE in case of OS/2 Warp 4.

For example, to install OS/2 Warp 4 on a ThinkPad 760ED (Line 36 in PCMCIA.TBL from OS/2 Warp 4 represents this ThinkPad), issue:

2. Modify CONFIG.SYS on Diskette 1

Insert Diskette 1 into Drive A: and make the following modifications to CONFIG.SYS:

- Comment out the line for the mouse driver:

```
REM device=\mouse.sys
```

- Change the viotbl line from:

```
devinfo=scr,vga,viotbl.dcp
```

to:

devinfo=scr,vga,VTBL850.dcp

- Erase unnecessary files from Diskette 2

Insert Diskette 2 into Drive A: and issue the following commands:

```
erase A:\UNPACK.EXE
erase A:\UNPACK2.EXE
erase A:\MOUSE.SYS
erase A:\VIOTBL.DCP
erase A:\RMVIEW.EXE
```

Erasing these files that are not used during the installation process frees up needed space on the boot diskette for TCP/IP and NFS.

3. Add TCP/IP support to the last boot diskette

Leave the last diskette in Drive A: and issue the following single-line command:

```
D:\CID\IMG\MPTS\THINLAPS.EXE D:\CID\IMG\MPTS A: IBMTOK.NIF /TCPIP
  /IP:x.xx.x.xxx
  /NM:xxx.xxx.xxx.x
  /H:xxxxxxxx
  /DMN:xxxxxxxxxxx.xxx.xxx
  /NSIP:x.xx.x.xxx
  /RT:x.xx.x.xxx
```

where:

- /IP The address that CLIENT1 will use.
- /NM Your TCP/IP net mask.
- /H The hostname that CLIENT1 will use.
- /DMN The name of your DNS domain.
- /NSIP The address of your DNS domain nameserver.
- /RT The address of your TCP/IP default router.

Note: The command consists of just one single line.

Consider Correct Ring Speed

Specify the correct NIF file for your adapter. For example, use IBMTOKCS.NIF for PCMCIA token-ring adapters.

You may need to edit the PROTOCOL.INI in these cases. For example, for IBMTOKCS.NIF, you may need to change the RINGSPEED keyword if the standard ring speed is not 16 MBit/sec.

4. Copy TCP/IP DLLs to image server

Insert Diskette 2 into drive A: and issue the following commands:

```
MD D:\CID\OS2WORK\SOFTDIST\TCPDLL
CD D:\CID\OS2WORK\SOFTDIST\TCPDLL
COPY A:\SO32DLL.DLL
COPY A:\TCP32DLL.DLL
PKUNZIP2 D:\CID\IMG\MPTS\MPTN\DLL\MDLL.ZIP . MPTN\DLL\TCPTIME.DLL
```

Note: Be sure to include the period parameter between .ZIP and MPTN.

5. Modify TCP/IP configuration from Diskette 2.

Insert Diskette 2 into drive A: and issue the following commands:

```
ERASE A:\SO32DLL.DLL
ERASE A:\TCP32DLL.DLL
COPY D:\CID\OS2WORK\OS2OPSYS\AFLEAN.SYS A:\AFINET.SYS
```

SO32DLL.DLL and TCP32DLL.DLL are not required on the boot diskettes and will be accessed from the image server after NFS has been started. AFLEAN.SYS provides a smaller version of the AFINET device driver, saving approximately 100 KB on the boot diskette.

Note: AFLEAN.SYS is available in the second MPTS Corrective Service Diskettes package, WR08415, as APAR IP20945. It is part of the ZIP file, MPROT2.ZIP, that resides in the diskettes's \MPTN\PROTOCOL directory.

6. Install NFS and the SD4OS2 client code on the last diskette

Leave the last diskette in drive A: and issue the following commands:

```
D:
CD D:\CID\OS2WORK\OS2OPSYS
MKOS2IP D:\CID\OS2WORK\OS2OPSYS D:\CID\IMG\OSWARP4 D:\TCPIP
```

where:

- D:\CID\OS2WORK\OS2OPSYS Is the directory for the pristine samples.
- D:\CID\IMG\OS2WARP4 Is the directory for OS/2 images.
- D:\TCP/IP Is the directory where the NFS kit is installed.

For OS/2 Warp 4, switch to Diskette 1 when prompted.

This batch file will install the NFS and SD4OS2 pristine client code on the pristine diskettes:

- Copy files from the OS2OPSYS samples:
 - FDISK.DAT, FNDVIP.CMD, FORMATC.DAT, FORMDSK.CMD, STARTIP.CMD, and PREPDSK.CMD
 - Copy files from the installed NFS kit and TCP/IP Applications:
 - MOUNT.EXE, NFS200.IFS, RPCDLL.DLL, and NFSCTL.EXE
 - Copy VDISK.SYS from the OS/2 images
 - Modify lines in CONFIG.SYS using DPFUPD:
 - Change the OS/2 shell line to read:
 - SET OS2_SHELL=CMD.EXE /K FNDVIP.CMD
 - Add . ; (period and semicolon) to the beginning of LIBPATH, PATH, and DPATH.
 - Add Z:\TCPDLL to the end of LIBPATH.
 - Add lines in CONFIG.SYS:

```
IFS=A:\NFS200.IFS
SET NFS.PERMISSION.BITS=750
SET NFS.PERMISSION.DBITS=750
DEVICE=\VDISK.SYS 1000, ,
```
- Note:** Instead of 750, you may use 700.

Target Specific Parameters

The SD4OS2 boot diskettes with TCP/IP support contain target-specific parameters (TCP/IP host name and TCP/IP address) and cannot be used unmodified concurrently on multiple targets. The TCP/IP protocol will detect an address conflict if a second system attempts to use the same address concurrently.

However, with some simple modifications, you can adapt these diskettes for concurrent pristine installs:

- Comment out the `MPTSTART` command in the Diskette 1's `CONFIG.SYS`:

```
REM CALL=CMD.EXE /Q /C A:\MPTSTART.CMD
```
- Change the `OS2SHELL` back to `CMD.EXE` in the Diskette 1's `CONFIG.SYS`:

```
SET OS2_SHELL=CMD.EXE
```
- Modify the TCP/IP initialization file `SETUP.CMD` on Diskette 1:

```
ifconfig lan0 %1 netmask xxx.xxx.xxx.x metric 0 mtu 1500
```

where `%1` replaces the hard-coded TCP/IP address and `xxx.xxx.xxx.x` is your site-specific TCP/IP net mask.

When you boot a target workstation with the modified set of diskettes, the boot process will stop at a command prompt, and you will need to issue these three additional commands:

1. Set the target's TCP/IP host name:

```
SET HOSTNAME=xxxxxxxx
```

where `xxxxxxxx` is your target's TCP/IP hostname.

2. Initialize the TCP/IP protocol support:

```
SETUP y.yy.y.yyy
```

where `y.yy.y.yyy` is your target's TCP/IP address.

3. Run the `VDISK` initialization procedure:

```
FNDVIP
```

The remainder of the pristine installation follows the standard process described in this document.

11.8.4 Creating Change Files

The profiles for each object to be installed at the target workstation have to be created at the software distribution server. You can find sample profiles in `D:\IMG\OS2WORK\OS2OPSYS`:

```

GLOBAL NAME: IBM.OS2WARP.REF.4000
DESCRIPTION: IBM OS/2 Warp 4.0 Pristine
CHANGE FILE TYPE: OS2CID
DEFAULT TOKEN: TargetDir=C:
INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\OS2WARP4\SEINST.EXE
PARAMETERS: /S:X:\IMG\OS2WARP4 /T:A:\ /B:${TargetDir}
            /R:X:\RSP\OS2WARP4\OS2.RSP
            /L1:X:\LOG\OS2WARP4\$(TARGET).L1

```

Figure 140. OS/2 Warp 4 Profile (SD4OS2) for Pristine Install: OS2.PRO

Note: The `PARAMETERS` line must be a single line.

```

GLOBAL NAME: IBM.MPTS.TCPIP.REF.1
DESCRIPTION: MPTS/TCPIP
DEFAULT TOKEN: TargetDir=C:
CHANGE FILE TYPE: OS2CID
INSTALL PROGRAM:
PROGRAM NAME: X:\OS2WORK\OS2OPSYS\MPTSIP.CMD
PARAMETERS: X:\IMG\MPTS\MPTS.EXE /E:MAINT /S:X:\IMG\MPTS
            /T:${TargetDir}\ /TU:${TargetDir}\ /R:${RSPFILE}
            /L1:X:\LOG\MPTS\$(TARGET).L1
RESPONSE FILE: X:\CID\OS2WORK\OS2OPSYS\MPTSIP.RSP

```

Figure 141. MPTS-TCP/IP Profile (SD4OS2): MPTSIP.PRO

Note: The `PARAMETERS` line must be a single line.

RESOLV2 Install Workaround

MPTS.EXE does not create RESOLV2 if one already exists in the path indicated by the `ETC` environment variable. Since the pristine install has a RESOLV2 file in the ETC directory on the VDISK, the standard install of MPTS would not create RESOLV2.

As a workaround, you invoke MPTSIP.CMD, which unsets `ETC`, and then invokes MPTS.EXE. Since the `ETC` environment variable is not set at that point, it correctly creates the RESOLV2 file on the hard disk.

```

GLOBAL NAME: IBM.TCPIP.NFS.REF.57064
DESCRIPTION: TCP/IP NFS ONLY Install
DEFAULT TOKEN: TargetDir=C:
CHANGE FILE TYPE: OS2CID
COMPRESSION TYPE: LZW
INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\NFS\TCPINST2.EXE
PARAMETERS: /A- /SF- /S:X:\IMG\NFS /T:$(TargetDir)\TCPIP
            /TU:$(TargetDir) /R:X:\RSP\NFS\NFS.RSP
            /L1:X:\LOG\NFS\$(TARGET).LOG

```

Figure 142. NFS Profile (SD4OS2): NFSPRIS.PRO

Note: The PARAMETERS line must be a single line.

```

GLOBAL NAME: IBM.SD2CLT.TCPPRIS.REF.1
DESCRIPTION: SD Client/2 TCP/IP Pristine Configuration
DEFAULT TOKEN: TargetDir=C:
CHANGE FILE TYPE: OS2CID
INSTALL PROGRAM:
PROGRAM NAME: X:\OS2WORK\OS2OPSYS\SDPRISIP.CMD
PARAMETERS: X: $(TargetDir) Z:

```

Figure 143. SD4OS2 Client Profile: SDPRISIP.PRO

Note: The PARAMETERS line must be a single line.

```

GLOBAL NAME: IBM.SD4OS2.CLTOP.REF.1
DESCRIPTION: SD4OS2 Client -- Over TCP/IP
CHANGE FILE TYPE: OS2CID
POST-INSTALL: DPFUPD.EXE /MOD:X:\RSP\SD4OS2\STARTUP.MOD
                /DRIVE:$(BOOTDRIVE)

INSTALL PROGRAM:
PROGRAM NAME: X:\IMG\SD4OS2\INSTALL.EXE
PARAMETERS: /S:X:\IMG\SD4OS2 /R:$(RSPFILE)
            /L1:X:\LOG\SD4OS2\$(TARGET).L1
            /L2:X:\LOG\SD4OS2\$(TARGET).L2 /X
RESPONSE FILE: D:\CID\OS2WORK\OS2OPSYS\SD4OS2IP.RSP

```

Figure 144. SD4OS2 Pristine Client Install over TCP/IP: SD4OS2IP.PRO

Note: The PARAMETERS line must be a single line.

To create the objects, execute the following commands:

```
D:
CD \CID\OS2WORK\OS2OPSYS
NVDM BLD WARP4.PRO
NVDM BLD MPTSIP.PRO
NVDM BLD NFSPRIS.PRO
NVDM BLD SDPRISIP.PRO
NVDM BLD SD4OS2IP.PRO
```

The objects are created and the global names added to the catalog.

11.9 Install the Pristine Client

This section illustrates steps executed to initiate a remote installation.

11.9.1 Define Client at SDSERV1

Issue the following command to define the pristine client:

```
NVDM ADDTG SDSCLT1 -s SDSCLT1 -a SDSCLT1 -y OS/2 -tp tcp:<hostname>
```

where:

<hostname> This is the TCP/IP hostname that you will use for SDSCLT1.

11.9.2 Define Shared Installation Parameters

Issue the following commands to add installation parameters:

```
NVDM ADDPM -a -i TCPNETMASK=xxx.xxx.xxx.x
NVDM ADDPM -a -i TCPROUTER=x.xx.x.xxx
NVDM ADDPM -a -i DNSDOMAIN=xxxxxxxxxx.xxx.xxx
NVDM ADDPM -a -i DNSADDRESS=x.xx.x.x
NVDM ADDPM -a -i SERVVERTCP=xxxxxxxx
```

where:

TCPNETMASK This is your TCP/IP net mask.

TCPROUTER This is the address of your TCP/IP default router.

DNSDOMAIN This is the name of your DNS domain.

DNSADDRESS This is the address of your DNS domain name server.
SERVERTCP This is the hostname that SDSERV1 will use.

11.9.3 Define Client Specific Installation Parameters

Execute the following commands:

```
NVDM ADDPM SDSCLT1 -i TCPADDRESS=x.xx.x.xxx  
NVDM ADDPM SDSCLT1 -i TCPHOSTNAME=xxxxxxxx
```

where:

TCPADDRESS This is the address that SDSCLT1 will use.
TCPHOSTNAME This is the host name that SDSCLT1 will use.

If you wish to install the pristine client on a drive other than C:, also define a value for TargetDir:

```
NVDM ADDPM SDSCLT1 -i TargetDir=D
```

Note: CLIENT1, by default, will install on the C: drive. You only need to specify this override token if you want to install it on another drive letter. Adding this token does not set up the Boot Manager settings that are needed to enable the boot of an alternate partition.

11.9.4 Submit Installation Commands at SDSERV1

Issue the following single-line commands to submit the installation:

```
NVDM INST IBM.OS2WARP.REF.1 IBM.MPTS.TCPIP.REF.1  
IBM.TCPIP.NFS.REF.57064  
IBM.SD2CLT.TCPPRIS.REF.1 -n -w CLIENT1  
NVDM INST IBM.SD4OS2.CLTIP.REF.1 -n -w CLIENT1
```

Note: The ⁿ flag indicates that this is a non-removable install. None of these OS2CID objects have BACKUP, REMOVE, or ACCEPT programs defined; so the installs must be non-removable.

11.9.5 Boot CLIENT1 from Diskettes

CLIENT1 can be booted from diskettes before or after submitting the installation commands:

1. Insert Diskette 0.
2. Reboot the system.
3. When prompted insert Diskette 1 and press **Enter**.

Normal Boot Message

You will see the following error messages as part of the normal boot process:

```
SYS0318: Message File OSO0001.MSG cannot be found for message 1467
```

SYS1467 is the VDISK initialization message

Occurs because OSO0001.MSG is not on the boot diskettes

```
SYS1804: The system cannot find the file SO32DLL
```

Occurs because the `INETWAIT` command attempts to load this DLL (and it is not on the diskette).

If you want to eliminate this message, comment `INETWAIT` out of `MPTSTART.COMD`.

4. OS/2 Warp 4 only: When prompted, insert Diskette 2 and press **Enter**.
5. Optional: Partition the hard disk.

Important

The hard disk must be partitioned, and the OS/2 installation drive must be formatted before starting the next step. These steps could have been done previously (because of a reinstall versus a printine install), or could be done manually.

6. From an OS/2 prompt, issue the following command:

```
STARTIP CLIENT1 CLIENT1 SDSERV1 SDSERV1 C
```

where:

`CLIENT1` This is the target name.

CLIENT1 This is the target address.
SDSERV1 This is the server name.
SDSERV1 This is the server address.
C: This is the target drive for the pristine installation.

11.9.6 Monitor Installation Results

The installation commands can be queued at the server before or after starting the pristine agent (through STARTIP.CMD). In either case, the pristine client will:

1. Execute STARTIP.CMD

The agent startup command performs various tasks:

- It establishes where to locate the client code.
- It establishes where the temp NVDM.CFG file it will use is located.
- It sets up a target directory for the install.
- It assigns the variables inputted at the command line.
- It starts the redirection.
- It creates a temporary NVDM.CFG file.
- It starts the software distribution.

2. When the install command has completed successfully, a screen similar to that below will be shown.

```
**Chosen Values**
(nvdm-client-name):
(nvdm-client-address max 8 chars):
(nvdm-server-name):
(nvdm-server-address max 12 char):
(target-drive-letter 1 char):
Building Service Directory...
Writing Software Distribution Configuration File...
Writing Software Distribution Final Configuration File...
Starting Software Distribution...
Connected to server <server_name>
Task fndcmps has pid<xx>
N/A
```

Figure 145. STARTIP.CMD Returned Messages When Run Successfully

3. The OS/2 installation will begin, displaying the standard blue installation screens.
4. The boot diskette can be removed after the *Loading System Files. Please wait....* blue installation screen gives way to the Copying Diskette x series of panels.
5. After the OS/2 install completes, the MPTS install will take place. MPTS writes it's messages to standard output and SD4OS2 redirects all the standard output to REQUEST.OUT; so you will not see any indication that the MPTS install is taking place.
6. After MPTS is installed, the NFS install will add client NFS support.
7. After NFS is installed, the pristine SD4OS2 client will be configured into the system by SDPRISIP.CMD.

This procedure updates the target's STARTUP.CMD to:

- Start the NFSCTL control program
 - Wait for NFS initialization to complete
 - Start the pristine agent
8. The second `NVDM INST` command will now begin. This command installs the SDS code on the pristine client and also removes the execution of FNDPRS from STARTUP.CMD (because it is no longer needed after the real agent is installed). This is done by running DPFUPD.EXE as a post-req along with this modification file.
 9. The install of SD4OS2 adds startup of the agent to the OS/2 startup folder; so it is started automatically on all subsequent reboots.
 10. The agent reboots, and the pristine installation is now complete.

11.9.7 Verify Request Completion

At SDSERV1, issue the following command:

```
NVDM LSRQ -w CLIENT1
```

to display the status of all requests for CLIENT1. The requests are completed successfully only if the Status shows as *Successful*:

```
Request ID: SDSERV1 root 445 0
Domain: SDSERV10
Target: CLIENT1
Submission time: 10/03/96 05:12 PM
Request type: Install
Object: IBM.OS2WARP.REF.1
Status: Successful
Error severity:0
Error code: 0000:0000
Completion time: 10/03/96 05:40 PM
Request ID: SDSERV1 root 446 0
Domain: SDSERV1
Target: CLIENT1
Submission time: 10/03/96 05:13 PM
Request type: Install
Object: IBM.SD4OS2.CLTOP.REF.1
Status: Successful
Error severity:0
Error code: 0000:0000
Completion time: 10/03/96 05:50 PM
```

Note that for a corequisite installation, only the first global name is shown in the Object field.

A completed request can be deleted by issuing the following command:

```
NVDM ERASEREQ xxx
```

where xxx is the numeric request number. The request numbers from the above examples are 445 and 446.

Appendix A. More Syntax Information and REXX Listings

This appendix provides syntax information for commands and utilities used in CID environments. It also displays a couple of REXX files used in all scenarios. Since some of the REXX files are huge, we put all REXX files on the CD-ROM that is packaged with this book rather than displaying them.

A.1 The SrvIFS Utility

The SrvIFS utility consists mainly of two parts:

- THINSRV

which installs the server function called SERVICE.EXE. At startup time SERVICE.EXE reads an initialization file called SERVICE.INI in which share names are defined.

- THINIFS

which provides the requester function. Certain parameters, such as directory assignments, have to be provided when the THINIFS command is executed.

A.1.1 The SERVICE.INI Keywords

The following is a description of the parameters used in the SrvIFS code server.INI file. The default configuration file is SERVICE.INI.

There can be a total of 11 setable parameters in the configuration file. Any line whose first character is one of the following is considered to be a comment.

- # Number sign
- ! Exclamation point
- @ At sign
- ; Semicolon
- * Asterisk

Blank lines are not permitted in a SrvIFS code server configuration file. Take care that there are no non-printable characters at the end of the file.

Name=<*server name*> Specifies the NetBIOS name of the SrvIFS code server.

This is the parameter that relates to the name

specified in a `SRVATTC` statement in the SrvIFS redirector's `CONFIG.SYS` file.

For SrvIFS, valid values are 1-15 alphanumeric characters.

Even though the SrvIFS server and client names are NetBIOS names, SrvIFS does not follow the NetBIOS naming convention. To lessen the confusion, we find it most practical that the name of the INI file is the same as the code server name defined in this parameter.

`GroupName = <YES / NO>`

Specifies whether the server's NetBIOS name is a group name or a unique name.

If `NO` then the server name must be unique on the network. The client workstation will request the use of a unique server. If `YES` then the server can be one of multiple code servers with the same name. These servers should provide the same service on a first come/first served basis by any client on the network.

Valid values are `YES` or `NO`.

It is a required parameter.

`Adapter = <0 / 1 / Both>`

Specifies the token-ring adapter used by the SrvIFS client server.

The server can support two adapters concurrently. Valid values are `0`, `1` or `Both`.

The default value is `0`.

`MaxClients = <number>`

Specifies the maximum number of concurrent active clients that will be allowed to connect to the server through each adapter.

If `Adapter=Both` is specified, this value applies to both network adapters.

Valid values are 1-100.

The default value is 1. Often this parameter needs to be increased.

MaxFiles = *<number>*

Specifies the maximum number of files that the server may have open concurrently.

This value should be at least as large as the number of concurrent clients that are expected to attach. Since installation programs often have multiple files open concurrently, a value equivalent to 3-4 files per concurrent client is suggested.

Valid values are 1-9999.

The default value is 100.

It has been found that some CID-enabled products may be opening 15 to 20 files at a time; so when you increase the number of your clients, you should also increase the value for MaxFiles.

ClientWorkers = *<number>* Specifies the number of threads used to support SrvIFS redirector's requests.

For small networks, a value of 6 is suggested. For larger networks, (20 or more concurrent SRVIFS redirector's) a value of 12 is recommended.

Valid values are 2-12.

The default value is 6.

Depending on the processor speed of your code server, the number of clients you want to install, and the LAN throughput, the value may need to be tuned.

The greater the number of clients the more you should increase the default value for

ClientWorkers.

Path = *<fully qualified path>*

Specifies the single, fully qualified path that will appear as the root of the redirected drive to the

SrvIFS redirectors.

This string does NOT contain a trailing backslash unless it is specifying the root directory of a specific drive.

Example:

Path = D:\CID

If a client makes a SRVATTCH X: CIDSRV

and the server name is CIDSRV then D:\CID will be available for the client as the root directory, X:. (If PerClient is set to NO.)

There is no default value.

PermitWrite = <YES / NO>

Specifies whether the clients can access the directory (and its subdirectories) specified in the PATH statement in Read/Write mode or Read-Only mode.

Valid values are YES (default) or NO.

PerClient = <YES / NO>

If this feature is enabled, a subdirectory descendant from the PATH= directory is used for each client.

The subdirectory name is the client name.

If a client REQ1 makes a SRVATTCH X: CIDSRV and PATH=D:\CID and PerClient is set to YES, then for this client, D:\CID\REQ1 will be made available as the root directory, X:.

Valid values are YES (default) or NO.

Alias = <ReadType, AccessType, Alias_name, Alias_path>

ReadType = <ReadOnly | ReadWrite>

Sets read or write access to Alias_path.

Valid values are ReadOnly or ReadWrite.

There is no default value.

AccessType = <Single | PerClient>

Single will cause Alias_path to be shared by ALL clients.

PerClient provides a unique view of the directory Alias_path by using the SrvIFS redirector's name as a subdirectory descendant of Alias_path. If subdirectory Alias_path\<SRVIFS redirector's name> does not exist; it will be created.

Valid values are Single or PerClient.

There is no default value.

Alias_name = <share name>Alias_name is the parameter that relates to the \\servername\Alias_name specified in a SrvAITCH statement in the SrvIFS redirector's CONFIG.SYS file. There, the server name corresponds to the value of Name parameter.

Valid values are 1-8 alphanumeric characters.

There is no default value.

Alias_path = <fully qualified path>

Fully qualified path used for Alias_name. If the directory specified as fully qualified path does not exist the server will not start.

There is no default value.

AuthList = <fully qualified path and file name>

This optional ASCII file is a list of authorized SrvIFS redirectors granted access to this SrvIFS code server.

There should be one line per client name in the form "Name.Address Comment" in this file. All comment markers described before are acceptable. Name is the SrvIFS redirector's name

specified in the IFS keyword statement of the SrvIFS redirector's CONFIG.SYS file. Name can optionally be followed by a Address and/or a Comment.

Valid values are 1-8 alphanumeric characters.

There is no default value.

Address is an optional LAN Universally Administered adapter address. For each SrvIFS redirector's entry in the `AuthList` file usage of the adapter address can restrict a SrvIFS redirector's access to a specific SrvIFS code server.

The address should be separated from the SRVIFS redirector's name by a period (.). No other characters, including spaces, may be included. It is also possible to specify an SrvIFS redirector's name as asterisk (*) followed by a LAN address to connect regardless of its SrvIFS redirector's name.

Valid value is 12 alphanumeric characters.

There is no default value.

Comment is separated from the Name or optionally the Address by one or more spaces. Valid values are alphanumeric characters.

A.1.2 Example Authorization List File for SERVICE.EXE

The following file applies to the optional `AuthList` parameter of the SrvIFS code server configuration file.

```
*      Start of Sample Authorization List
*
#
CLIENT1          SRVIFS redirector's
!                name
!
CLIENT2.A1F6955A0010 SRVIFS redirector's name
!                with address supplied
*.100005876543   Wildcard SRVIFS redirector's
!                name with address supplied
#
*      End of Sample Authorization List
```

Figure 146. Example of an SrvIFS Authorization List

If the authorization list file is not found, an error message will be displayed, and the SrvIFS code server is not started. Invalid SrvIFS redirector's names are ignored. The invalid name will be displayed when the SrvIFS code server is started. The server will continue to run. Attempted access by an unauthorized SrvIFS redirector will result in the following message:

Connection to server disk is rejected.

A.1.3 Example SERVICE.INI File

The following figure displays a sample SERVICE.INI file.

```

* SERVICE.INI file used by SERVICE.EXE
#
Adapter = 0      * Number of token-ring adapter being used
MaxClients=25   * Adapter 0 is set to support up to 25 SrvIFS
                * redirectors
MaxFiles = 102  * Maximum number of concurrently opened files is set
                * to 102
Name=CIDSRV     * Name of the code server
GroupName=No    * NO means that the code server name must be unique
ClientWorkers=12 * Number of threads used to support SRVIFS
                * redirectors' requests
Path=D:\CID     * full path to the default SRVIFS code server
                * directory

PerClient=No
PermitWrite = No
alias= ReadOnly,Single,img,D:\CID
alias= ReadWrite,Single,log,D:\CID\LOG
alias= ReadOnly,Single,tools,D:\OS2TOOLS

```

Figure 147. Example of SERVICE.INI

A.2 Autopartitioning the Hard Disk

Before you install the operation system on the hard file, you have to partition the hard drive. The easiest way to do this is to call the `FDISK` command with required parameters as the first step of the installation process. `FDISK`ing the hard disk would be wise when reinstalling or upgrading an existing system and when the previous configuration does not need to be migrated.

After partitioning, the partitions have to formatted. You can use the `FORMAT` command, or you add an entry to the OS/2 response file to format the partition during the installation process.

When using the Boot Manager option with several operating systems residing on the same workstation, the following restrictions apply:

- DOS
 - Must reside on first primary partition of the first physical disk. If the DOS version is earlier than 4.0, this partition can't be greater than 32 MB.
- Windows NT

Windows NT loader must reside on the primary C-partition. The operating system itself can be installed on another drive.

- Windows 95

Windows 95 can boot only from the primary C-partition, but the operating system itself can be installed on another drive. As far as file systems and sharing of logical disks are concerned, be aware of the VFAT file system that Windows 95 installs. VFAT is not compatible with FAT.

- Boot Manager

Must reside in the first 1024 cylinders of the first hard disk if the system BIOS does not support LBA mode. Typically, the first 1024 cylinders is equal to 512 MB. All the new machine BIOSes support more than 1024 cylinder IDE hard disks, and Boot Manager can be installed above the 1024 cylinder limit.

Note

The number of partitions on one hard disk is limited. The maximum number of primary and extended partitions on a single disk is four, including Boot Manager. The maximum number of logical partitions in the extended partition is eight. If you install Boot Manager and three other primary partitions, there is no room for extended partition anymore.

A.2.1 The Fixed Disk Utility Program (FDISK)

The `FDISK` command provides functions such as creation of partitions or logical drives, their deletion and/or making them startable, all of which are needed to make logical drives accessible for data and programs.

There are two types of FDISK programs, each providing the same functions:

- FDISK

This program has a VIO interface because it is used in several environments where PM is not available. It accepts command line parameters for use in scripts.

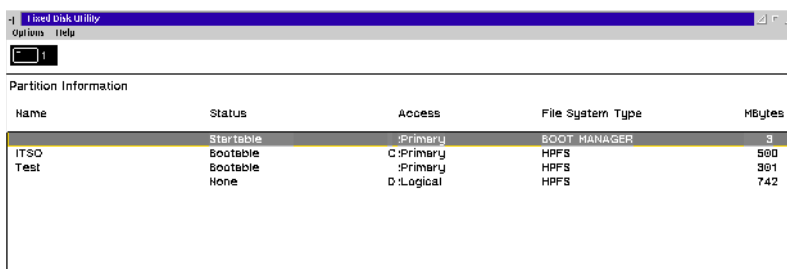
- FDISKPM for OS/2 Presentation Manager

Gives the same functionality as command line FDISK, but the interface is a user-friendly graphical interface.

A.2.2 Functionality of FDISK

FDISK for OS/2 enables the user to partition the hard disks and install Boot Manager to support booting of multiple operating systems on the same system.

The FDISK screen has five columns containing specific information about the partitions that exist on the system hard disk. Each hard disk is represented by a number at the top of the FDISK screen:



The screenshot shows a window titled "Fixed Disk Utility" with a menu bar containing "Options" and "Help". Below the menu bar is a "Partition Information" section. It contains a table with five columns: Name, Status, Access, File System Type, and MBytes. The table lists three partitions: ITSD (Startable, :Primary, HPFS, 500), Test (Bootable, :Primary, HPFS, 301), and Test (None, D:Logical, HPFS, 742).

Name	Status	Access	File System Type	MBytes
ITSD	Startable	:Primary	BOOT MANAGER	500
Test	Bootable	:Primary	HPFS	301
Test	None	D:Logical	HPFS	742

Figure 148. FDISKPM Window

When you select a hard disk, its partition information is displayed in the window. Partitions are either primary partitions or logical drives within an extended partition. Any free space (space within the hard disk that is available for more partitions) is also displayed in the window.

This information includes:

Name This is the name that has been assigned to any primary partition or logical drive to be displayed on the Boot Manager menu.

Status Indicates if a partition is Installable, Bootable, Startable or none of the above.

If set to Installable, the respective partition will be used as the target for continuing the OS/2 install.

If set to Bootable, the respective partition is displayed on the Boot Manager menu when the system is restarted.

If set to Startable, the system restarts directly from this partition and will be Installable.

Remember: One of the primary partitions must be set to Startable for the system to restart successfully.

Access If a partition is accessible, this column indicates if the partition is a primary partition or a logical drive within the extended partition and the drive letter assigned to the partition.

Only one primary partition is accessible at a time on each physical drive.

FS Type Indicates the type of file system on the partition. Any partitions that have not been formatted will be displayed as Unformatted.

Any area on the hard disk not assigned to a partition will be displayed as Free Space.

MBytes Indicates the size in megabytes of the partition or free space.

To modify a disk setup, select the **partition** or **Free Space** entry on the FDISK screen and then press **Enter** (or use mouse in FDISKPM) to open the Options pull-down menu. Use the tab key to activate the disk selection.

11.9.7.1 Functions on Options Pull-Down Menu of FDISK

The following functions are available on the Options pull-down menu of FDISK.

Install Boot Manager	Creation of Boot Manager partition and loading the Boot Manager program.
Create Partition...	Creation of primary or logical drive.
Add to Boot Manager menu...	Makes a partition bootable and adds it to the Boot Manager menu.
Change Partition Name...	Change name of partition in Boot Manager menu.
Assign C: Partition	In case of multiple primary partition, selection of default partition in the Boot Manager menu.
Set Startup Values...	Specify startup values such as a default partition, startup menu timeout, or mode for the startup menu.
Remove from Boot Manager menu	Removes the partition from the Boot Manager menu and makes the partition non-startable.
Delete Partition	Deletes the selected partition.

Set Installable	Sets the selected partition as installation partition.
Make Startable	Use this choice to set a primary partition as the direct restart target. For Boot Manager support, the Boot Manager partition must be set to Startable.

Note

If you change partition table by creating new partitions or removing partitions, you are forced to reboot the system to make the changes effective.

A.2.3 FDISK Command Line Interface

The `FDISK` command line interface can be used attended from the OS/2 command line or called unattended from a REXX script or from a `.CMD` file. `FDISK` syntax and parameters are described in the following section in detail.

FDISK Syntax

```
FDISK </command:value /restrictor:value>
```

The command parameter initiates the actual execution of the command. The following commands are available:

- `/CREATE[:name]` Creates a partition, optional a name can be assigned, that appears in the Boot Manager menu.
- `/DELETE[:all]` Deletes a partition. To delete all partitions on a physical disk, use the command `FDISK /DELETE:ALL /DISK:n`, where `n` is the disk number.

Note

Be careful with the use of `/DELETE:all` because on some PS/2 machines, the reference partition is not hidden, and it will be deleted by this command unless you specify a `FSTYPE` mask in the restriction parameter.

Deleting a partition removes all the data from the disk.

- `/FILE:<filename>` Processes all `FDISK` commands in the file `<filename>`.
- `/NEWMBR` Rewrites the Master Boot Record information.

/QUERY Returns the list of the current partition setup; the output looks like follows:

Drive Name	Partition	Vtype	FStype	Status	Start	Size
1 0000003F	:	1	0a	2	0	3
1 ITSO	C:	1	07	1	3	500
1 Test	:	1	17	1	504	301
1 001929ff	D:	2	07	0	805	742

The /QUERY parameter can be used in conjunction with other parameters to filter the output. The parameters used are listed in Table 21. For example, to list all the primary partitions on the disk one, the command used would be: `FDISK /QUERY /VTYPE:1 /DISK:1`. and the command `FDISK /QUERY /DISK:2 /VTYPE:2` would display all the logical partitions on the disk number two.

Table 21. Keywords for FDISK /QUERY (Part 1 of 2)

Keyword	Value (hex)	Meaning
Vtype	1	primary partition
	2	logical partition
Status	0	none
	1	bootable
	2	startable
FStype	1	FAT less than 16 MB
	2	XENIX
	4	FAT 16 MB to 32 MB
	5	Extended
	6	FAT more then 32 MB

Table 22. Keywords for FDISK/QUERY (Part 2 of 2)

Keyword	Value (hex)	Meaning
	7	HPFS
	8	AIX bootable
	9	AIX non-bootable
	a	Boot Manager
	11	hidden less then 16 MB
	12	XENIX hidden
	14	extended hidden
	16	unknown
	82	Solaris

Note

If any *****BIOS xxx** statements are displayed from the **FDISK** query, this indicates that the system hardware BIOS can only see a limited amount of the hard drive space available. The number after the *****BIOS** indicates how many megabytes of drive space the BIOS can see. Any partition that is not inside of this range will not be seen by the BIOS.

/SETACCESS Use this setting to select a primary partition as the C: drive if you have multiple primary partitions.

/SETNAME: <newname>

Sets the boot name of partition and makes it bootable from the Boot Manager menu. If the new name is left blank, the boot name is removed and will not be selectable from the Boot Manager menu.

/STARTABLE Sets a partition startable, selects the default partition to boot automatically. The OS/2 installation process will automatically set the Boot Manager partition startable, if one is present.

If you set some other than Boot Manager partition to be startable, the Boot Manager will be disabled.

Contrary to commands, restrictors are arguments that limit the actions of the commands. For example, the command `QUERY`, without any restrictors, would output all partitions on all drives in the system.

If the restrictor `/DISK:2 /VTYPE:1` is added, then only the primary partitions on drive 2 would be displayed.

The following restrictors and associated values are available:

`/BOOTABLE:s` Specifies the "boot selectable" status where `s` can be:
0 = specifies non-bootable partitions
1 = specifies bootable partitions

`/BOOTMGR` Specifies the Boot Manager partition.

`/DISK:n` Specifies the disk number.

`/FSTYPE:hxx (or) :tttttt`

File system type:

`hxx` = where `xx` is the system indicator as defined in the partition table or `tttttt` can be:

dos
fat
hpfs
free
other

`/NAME:ccccccc` Selects a partition with a specific name.

When doing a query operation, a pseudo name is assigned to every partition and free space that doesn't have a boot name assigned.

Note

This name is not set as the partition name, but is only used as a temporary identifier for the user.

`/SIZE:mmmm` The size of the partition where `mmmm` is the amount of space in megabytes.

`/START:c` Used when creating a partition starting at location `c`, where `c` can be:

t = top of partition
b = bottom of partition

`/SYSTEM` Can be used to create a system partition with a bootload on it or as a selecton parameter.

`/VOLID` Is used to select a partition with a particular volume ID number.

This is like the `/NAME` parameter, but is used when no name has been assigned to a particular partition.

This is a new function for OS/2 Warp 4.

`/VTYPE:X` used to define the partition type with the `/CREATE` parameter, or can be used as restriction parameter to select a sepecific partion.

The value of `x` can be:

- 0 = unusable
- 1 = primary
- 2 = logical
- 3 = primary or logical

A.2.4 FDISK ReturnCodes

The `FDISK` command gives an return code for the operation done. The `FDISK` return code list represented in Table 23 includes only some code values, but should cover the most common error situations.

Table 23. Some FDISK Return Codes

Return Code (Dec)	Return Code (Hex)	Explanation
0	0	Operation successful
1	1	Operation unsuccessful
5120	1400	Operation successful
7168	1C00	Successfully created a Boot Manager partition
7680	1E00	Successfully deleted all partitions
6144	1800	Successfully deleted Boot Manager partition

A.2.5 Remarks

The PC system BIOS provides the access to the actual hardware including the hard disk. The BIOS gains access using the INT 13, which has only 10 bits available to address a cylinder on the hard disk. This limits the INT 13 accessible hard disk to 1024 cylinders or less.

Because the new hard drive capacity requirements have grown, there was a need to find round-a-way to the 1024 cylinder addressing problem. The solution used is called LBA mode (Logical Block Addressing) for IDE-type hard drives. LBA typically divides the actual cylinder count by two and multiplies the head count by two. LBA then makes the conversion, when a read or write operation is requested, from the logical values to the real, physical values.

Note

If you partition the hard disk when the LBA mode is disabled, the Boot Manager and all the bootable partitions have to be inside the first 1024 cylinder area. OS/2 and Windows NT can access logical partitions above the limit, unlike DOS/Windows.

If you later on want to enable the LBA mode, you have to partition the whole hard disk again.

A.2.6 The FDISK /FILE Parameter

FDISK operations to be executed can be stored in a response file and use the /FILE option to pass the input file to the command. Using this parameter makes it easy to delete and create multiple partitions in the unattended mode. The response file for FDISK is an ASCII file where FDISK operations are entered one per line and parameters are separated by a comma. In other words: FDISK reads the response file row by row and parses each row as command line parameters.

For example, to delete all the partitions on the disk one and create maintenance, system and data partitions after erasing the partitions, the commands executed would be:

```
FDISK /FILE:DELETE.DAT
FDISK /CREATE /BOOTMGR /DISK:1
FDISK /FILE:CREATE.DAT
```

The operation has to be completed in three phases: first the deletion of all the partitions, then Boot Manager creation and at last the creation of new partitions.

Here is the sample DELETE.DAT file to delete all the partitions on disk number 1:

```
/delete:all,/disk:1
```

The Boot Manager has to be installed in a separate phase because the partition information is updated only after completing all the operation in the response file.

When the Boot Manager is created, all the other partitions can be created. Here is the sample CREATE.DAT file to create a primary partition of 500 MB for the system in the start of the free space, maintenance partition of 40 MB in the end of the free space using vtype 2, which will create a logical partition, and finally the data partition using all the free space left on disk one.

```
/create:WARP4,/size:500,/disk:1,/vtype:1,/start:t  
/create:MAINT,/size:40,/disk:1,/vtype:2,/start:b  
/create,/disk:1,/vtype:2
```

These operations could be completed using the normal command line parameters, but using the `/FILE` parameter makes it easier to change partition information without changing the command file.

Note

The `FDISK` maximum command line length is limited. If you enter too long a command line or define too many operations to a response file, one or more of the parameters may be ignored.

A.3 SETBOOT

The `SETBOOT` command is used to control the Boot Manager and allows you to use some of the functionality provided by the Boot Manager.

SETBOOT Syntax

```
SETBOOT <option>
```

The following options of `SETBOOT` are very useful for CID installations:

`/B` Performs proper shutdown reboot the system and displays the Boot Manager menu.

`/IBD:d` Performs shutdown and restarts with the operating system, that resides on drive d. Bootmanager will not be shown.

For example, `SETBOOT /IBD:C` would reboot the workstation form drive C:.

`/IBA:<name>` Performs proper shutdown and bootup on drive with the name specified. If you have a system with a name that includes space character, you have to use quote marks.

For example, `SETBOOT /IBA:OS2V4` would restart you workstation from partition named OS2V4.

`/n:<name>` Sets the partition specified by te alias name and its corresponding index value as the operating system to be started. Up to four pair-combinations can be given, one for each index value, 0 to 3.

`n = 0` assigns the specified alias as the default operating system.

`n = 1 to 3` specifies the alias to be started when the corresponding index is chosen to start.

To define the default partition to be booted:
`SETBOOT /0:OS2V4`

`/X:x` Sets the system startup index to indicate the partition to be started. The value x can be set from 0 to 3.

A value of 0 sets the Boot Manager to attended mode and provides for a default system selection.

A value of 1 to 3 in this index puts the Boot Manager in unattended mode.

If a system fails to start the system set by the index (see `/n:name` for setting the index), the subsequent startup would try to start as fallback system, and so on.

If you do not set this value before you start the system again, the Boot Manager will start a different partition.

To put Boot Manager in attended mode use this command:

```
SETBOOT /X:0
```

To set the startup index to put the Boot Manager in unattended mode, enter this command:

```
SETBOOT /X:3
```

/Q

Queries the setting of the Boot Manger.

/T:x

Sets the timeout delay for the display of the Boot Manager menu. The Boot Manager menu can be disabled by setting the value to 0.

If you specify /T:NO then then the Boot Manager menu will have not timeout value and will stay on the screen until a selection is made.

A.4 Partitioning Utility: DISKPRP

DISKPRP.CMD is a REXX procedure to partition the hard disk. It controls if the disk has to be partitioned, and if so it uses FDISK (see "The Fixed Disk Utility Program (FDISK)" on page 449) to do it.

This is how the procedure works:

- Partitioning is done when no partition with the name of WARP can be found.
- If it finds a partition with name of WARP it will go ahead and format the partition.
- To force partitioning, the procedure reads first the `PARTITION` environment variable. If it is set to `YES`, the procedure deletes first all the existing partitions on disk.

Setting the environment variable can be done in several ways:

1. Modifying the `CONFIG.SYS` file to introduce a command

```
SET PARTITION=YES
```

2. Typing the command at the command prompt before executing `STARTUP.CMD`.
3. Modifying `STARTUP.CMD` to introduce the command in the file.

If CONFIG.SYS or STARTUP.CMD have been modified, they have to be restored to their original content to be used again.

A way to do that is to copy the original diskette STARTUP.CMD file to STARTUP.NO and create a STARTUP.YES file with the variable set. Uncomment in DISKPRP.CMD the line that copies STARTUP.NO to STARTUP.CMD.

Before booting on the diskettes, insert the diskette with STARTUP.CMD in any machine, and at a command prompt, copy STARTUP.YES to STARTUP.CMD.

```

/*****
/* This program performs the following steps - */
/*
/* 1.Looks for an environment variable PARTITION */
/* If set to "YES" then deletes partitions */
/*
/* 2.Queries disk partitions for a partition called */
/* "diskname" */
/*
/* 3.If none with that name are found - */
/* 2a. Execute FDISK */
/* 2b. Automatically reboots the workstation */
/*
/* 4.If a partition called "diskname" exists */
/* program exits. */
/*
/*
/*****
address cmd
'@echo off'
/*-----Diskname to look for -----*/
diskname='WARP'
os = 'warp40'
spath='x:\cid'
rdir='1>nul 2>nul'
img=spath'\img\'||os
fdisk=img'\DISK_2\FDISK'

/* if environment variable PARTITION set to YES : delete previous partitions */
if value('PARTITION', 'OS2ENVIRONMENT') = 'YES' then do
fdisk' /delete:all /disk:1 ' rdir
/* copy a:\startup.no a:\startup.cmd */
end

/* query the disk and queue the output */
fdisk' /query | 'spath'\rexx\RXQUEUE'
doneit=0 /* reset doneit flag */

do queued()
parse pull drive name part vtype fstype status start size
if name=diskname then doneit=1
end
if doneit=0 then
do
/* Execute the disk partitioning */
/* Assume 1 physical disk */

```

```

fdisk' /delete:all /disk:1 ' rdir
searchd='1'
fdisk' /query | 'spath'\rexx\RXQUEUE'
do queued()
parse pull drive name part vtype fstype status start size
if drive=searchd
then do
x=size
select
when x<1000 then do /* size of drive c: equal to x */

/* Add BootMgr */
fdisk' /create /bootmgr ' rdir

buff = '/create:'diskname' /vtype:1 /disk:1 /bootable:1'
fdisk buff' 'rdir
/*Boot the workstation from drive C:*/
call BANNER
end

otherwise do /* size of drive c: limited to 400k */
/* size of drive d: space left */
/* Add BootMgr */
fdisk' /create /bootmgr ' rdir

buff = '/create:'diskname' /size:400 /vtype:1 /disk:1 /bootable:1'
fdisk buff' 'rdir
/*Boot the workstation from drive C:*/
buff = '/create /vtype:2 /disk:1 '
fdisk buff' 'rdir
call BANNER
end
end
end
end
end
else exit

BANNER:
/*****
/* Subroutine to write banner to screen */
*****/
say '[0;7m';
do; /* Normal information message */
'@cls'; /* Clear the screen */
say '[8;1H'; /* Move cursor to line 8 */
say ' +' || left(' ',76,'+') || '+'; /* Put */
say '| ' || left(' ',76) || '|'; /* message */
say '| ' || center('The fixed disk preparation is complete.',76)|| '|';
say '| ' || left(' ',76) || '|';
say '| ' || center('The System must now be rebooted.',76) || '|';
say '| ' || left(' ',76) || '|';
say '| ' || center('Remove diskette from drive a:',76) || '|';
say '| ' || center('and insert the OS/2 V2 Install diskette.',76) || '|';
say '| ' || left(' ',76) || '|';
say '| ' || center('If using Remote IPL (from diskette or RIPL chip)',76) || '|';
say '| ' || center('do NOT do anything except press any key.',76) || '|';
say '| ' || left(' ',76) || '|';
say '| ' || center(' Press any key ',76) || '|';
say '| ' || left(' ',76) || '|'; /* in a */

```

```

say ' +' || left('+',76,'+') || '+'; /* pretty box */
end;
'Pause'
spath'\exe\os2\setboot /IBD:C'

```

A.5 The Desktop Shuffler Tool

The Desktop Shuffler, `CLNDESK`, lets you move, delete, or add objects to an OS/2 Warp 4 desktop in attended and unattended mode. You can issue the `CLNDESK` command at the actual desktop to be modified, or you can issue the command at one central donor system as part of building a specific desktop to be downloaded to other systems. No programming or REXX utilities are required.

Read about the sequence of using the migration tools before you use the shuffler.

The `CLNDESK` command reads the list of actions from a control file and from a keywords file. You can create, copy, move, shadow, delete, and modify objects.

A.5.1 CLNDESK Syntax

The syntax of the `CLNDESK` command is as follows:

CLNDESK Syntax

```
<x:\path>CLNDESK <control_file> <keywords_file>
```

where:

`<control_file>` Required.

Contains actions to take on the specified objects.

`<keywords_file>` Required.

Contains blank-delimited keywords of the form:

```
products=keyword1 [ keyword2 keyword2 keyword2]
```

Example:

```
C:\IBMINST\CLNDESK shuffle_ctl.lst shuffle_key.lst
```

A.5.2 Return Codes

The Desktop Shuffler utility can return one of the following codes. The codes are consistent with these in the *CID Enablement Guidelines*, S10H-9666.

- 0 Successful program termination, no reboot required.
- 4 Successful program termination, warning messages logged; no reboot required.
- 5 Unsuccessful program termination, no reboot required. Error messages logged.
- 6 Unsuccessful program termination, no reboot required.

Unsuccessful program termination with unexpected internal errors detected and logged.

A.5.3 Control File Syntax

You can use the Desktop Shuffler with the supplied sample control file to set the desktop objects for an OS/2 Warp 4 system to the same arrangement as an attended-installed system, or you can read the rules and examples below to create your own file.

Follow the general rules below when creating or editing a control file. Each section has any additional rules or options that apply to that section.

The control file is an ASCII file in typical .INI format. All the file sections are required, but each section can be empty.

Lines beginning with a semicolon (;) are comment lines. Blank lines are allowed. Each item on a line can be separated by zero or more spaces. Actions fail if objects referred to (except for create) do not exist. Section titles, for example [create], are not case sensitive.

Sections in the control file are executed in the following order:

1. [System]
2. [Create]
3. [Delete]
4. [Update]
5. [Move]
6. [Shadow]
7. [Copy]

8. [Update]

A created object is created in the hidden folder, updated, and then moved or shadowed to its appropriate locations.

Use `#if` condition, `#endif` condition, and `#else` condition directives in the keywords file to control execution.

A.5.3.1 Error Reporting from the Control File

If a syntax error is detected in the control file, an error message is issued to standard error.

A.5.3.2 Sample Control File

This annotated sample is a typical control file. It may or may not work; it is just for illustration.

```
; Shuffler Sample Control File
;
; a line starting with a semicolon is a comment line.
; This control file sets the desktop objects for a
; OS/2 Warp 4 system to the same arrangements as an attended
; installed OS/2 Warp 4 system.
;
; Note: The contents of the keyword files to be used
; with this example are:

Products=WARP OS2PEER TCPIP MPTS LDR
;
```

A.5.3.3 Section Syntax Information

The following options exist for control file keywords:

[System] This section can be either empty or all comment lines.

[Create] *Object Class*, *<ObjectID>*, *"Title"*, *Option*, *<Location>*

There can be 0 or more spaces before or after each item.

Object Class must be valid Workplace Shell class.

<ObjectID> must be a valid Workplace Shell *Object ID* name.

You must specify the brackets, for instance *<* and *>*.

"Title" is a string surrounded by double quotation marks. Use

a carat (^) to insert a new line into the object title displayed to the user.

Option can be:

FAIL	The object is not created if another object with the same <i>object_id</i> exists. FAIL must be uppercase.
REPLACE	The new object replaces an existing object with the same <i>object_id</i> . REPLACE must be uppercase.
UPDATE	The new object is not created if another object with the same <i>object_id</i> exists. The corresponding attributes are set to those of the new object. UPDATE must be uppercase.

One or more WPClass setup-strings (keyname/value pairs) can follow the option. See the *Workplace Shell Programming Reference* for additional information.

Location can be the *object_id* of any folder or the desktop <WP_DESKTOP>.

Sample create section:

```
[create]
; this will create a folder object
WPFolder, <MY_Folder>, "This is a title", REPLACE
; this will not create a folder object
WPFolder, <MY_Folder>, "This is a title", FAIL
; this will update the object with a new title
WPFolder, <MY_Folder>, "This is an updated title", UPDATE

[Delete]      ; The next two lines each delete one object
               <CPPVISBLD>
               <CPPIPMD>

[Move]        <ObjectID>, <Destination ObjectID>
```

An error is generated if both objects do not exist.

The object is not moved if it already exists in the destination object.

Example:

```
[move]
;The next line moves <CPPVISBLD> to the desktop
<CPPVISBLD> <WP_DESKTOP>
```

[Shadow]

<ObjectID>, *<Destination ObjectID>*

An error is generated if both objects do not exist.

The object is not shadowed if it already exists in the destination object.

The resulting object's ID is the original ID with "_SHADOW" appended to it.

When the original object is deleted, all its shadows are removed simultaneously.

If an object ID of the name *<object_id_SHADOW>* already exists in the destination folder, a new shadow is not created.

Example:

```
[shadow]
;The next line shadows <CPPVISBLD> from its current
; location to the desktop
; The shadow's Object ID will be <CPPVISBLD_SHADOW>
<CPPVISBLD> <WP_DESKTOP>
```

[Copy]

<ObjectID>, *<Destination ObjectID>*

An error is generated if both objects do not exist.

The object is not copied if it already exists in the destination object

The resulting object's ID is the original ID with "_COPY" appended to it.

When the original object is deleted, all its copies are unaffected.

If an object ID of the name *<object_id_COPY>* already exists in the destination folder, a new copy is not created.

Example:

```
[copy]
; The next line copies <CPPVISBLD> from its current
```

```
; location to the desktop
; The copy's Object ID will be <CPPVISBLD_COPY>
<CPPVISBLD> <WP_DESKTOP>
```

[Update] 0 or more lines with one of the following formats:

```
<ObjectID>, keyword = value
FOLDER<ObjectID>, keyword = value ; keyword = value
SETUPSTRING<ObjectID>, setupString
```

Separate multiple keyword = value pairs with semicolons (;).

See the *Workplace Shell Programming Reference* for information about *setupString*.

For TITLE=value pairs, do not use quotation marks around the title string.

Example:

```
[Update]
SETUPSTRING <WP_INFO>, TITLE=Utilities; DEFAULTVIEW=TREE
; connections folder
FOLDER <WP_CONNECTIONS> DEFAULTVIEW = TREE
FOLDER <WP_CONNECTIONS> TREEVIEW = MINI,LINES
; rename Productivity Folder to Utilities
<WP_TOOLS> TITLE = Utilities
; set information folder to treeview
<WP_INFO> DEFAULTVIEW = TREE
;
; end of sample control file
```

Keywords for Use in Control File [Create] and [Update] Options

Supported keywords for non-FOLDER entries correspond to the following subset of `wpSetup` override key/values supported by the `WPObj` class. See the *Workplace Object Classes* section, *WPObj* subsection of the *Workplace Shell Programming Reference* for additional information, or see a table of the keyname=value pairs.

```
CCVIEW=DEFAULT | YES | NO
DEFAULTVIEW = SETTINGS | DEFAULT | 0-9
ICONFILE = iconfile.ICO
ICONPOS = x,y //values between 0 and 100
MINWIN = HIDE | VIEWER | DESKTOP
NOCOPY = YES | NO
```

```
NODELETE = YES | NO
NODRAG = YES | NO
NODROP = YES | NO
NOLINK = YES | NO
NOMOVE = YES | NO
NOPRINT = YES | NO
NORENAME = YES | NO
NOSETTINGS = YES | NO
NOSHADOW = YES | NO
NOTVISIBLE = YES | NO
OPEN = SETTINGS DEFAULT
TEMPLATE = YES | NO
TITLE = Title_string
```

There is a lot of helpful information in the online information for the OS/2 Toolkit in the \toolkit\toolkit\books directory.

A.5.4 Errors and Error Recovery

The Desktop Shuffler utility ignores most lines in error. If the utility cannot interpret a line or if the line is missing a comma, the action is not done. The following lines produce no action:

Example 1:

```
[create]
WPFolder <R_FOLDER> "Z folder" FAIL, <Other_APP>
```

If the section titles are missing or out of order, the clndesk.log file contains the entry:

```
CCL0011: key not contained
```

Example 2:

```
C:\IBMINST\CLNDESK.EXE
```

If the keywords file is missing, an error is displayed in a dialog box, and the CLNDESK.EXE file is not created.

A.5.5 Limitations and Resitricitions

The shuffler utility cannot perform actions on the Warp Center desktop object. For instance, no folder or program objects can be copied to the Warp Center. Also, no WPS objects already in the Warp Center can be updated by the shuffler utility. For example, the following shuffler control file line produces no action:

```
[copy]
```

<WP_EPM>, <WP_WARPCENTER>

A.5.6 The SHUF_CTL.INI Control File

The SHUF_CTL.INI file includes all actions the Desktop Shuffler has to perform to rearrange all components of OS/2 Warp 4. The second input file, SHUF_KEY.INI, specifies the components installed that need to be rearranged.

```
;
;
; This control file sets the Desktop objects for a CID-installed
; OS/2 Warp 4 system to the same
; arrangements as an attended installed system
;
;
[system]
;
;
[create]
;
;
; Networking product install icons
; "Selective Install^for Networking"
WPPprogram, <CLTSTART>, "Selective Install^for Networking", REPLACE, <WP_INSTREMFOLDER>,
EXENAME=\IBMINST\NPCONFIG.EXE;ICONFILE=\IBMINST\ICONS\1091.ICO
; "Remove Installation^for Networking"
WPPprogram, <CLTRMOVE>, "Remove Installation^for Networking" ,
REPLACE, <WP_INSTREMFOLDER>,
EXENAME=\IBMINST\NPRMINST.EXE;ICONFILE=\IBMINST\ICONS\1093.ICO
; "OS/2 Warp^Remote Install"
WPPprogram, <CLT_REMOTE>, "OS/2 Warp^Remote Install", REPLACE, <WP_INSTREMFOLDER>,
EXENAME=\IBMINST\NPSETUP.EXE;ICONFILE=\IBMINST\ICONS\1090.ICO
;
;
;
;
; #if MPTS
; Create MPTS^Network Adapters^and Protocol Services
WPPprogram, <NTS_MPTS_ICON>, "MPTS^Network Adapters^and Protocol Services", UPDATE,
<WP_CONFIG>, EXENAME=MPTS.EXE
; Create DHCP Monitor
WPPprogram, <WS_DHCP_MONITOR>, "DHCP Monitor", UPDATE, <WP_CONFIG>
; Create DDNS Configuration
WPPprogram, <MPTS_DDNS_OBJECT_SETUP>, "DDNS Configuration", UPDATE, <WP_CONFIG>
; Create MPTS Guide
WPPprogram, <MPTSCfg>, "MPTS Guide", REPLACE, <WP_TASKSINFO>
; Create TCP/IP DHCP^Client Administration Guide
WPPprogram, <NP_DHCP_GUIDE>, "TCP/IP DHCP^Client Administration Guide", REPLACE,
<WP_TASKSINFO>
; Create TCP/IP Dynamic IP Introduction
WPPprogram, <NP_DIP_INTRO>, "TCP/IP Dynamic IP Introduction", REPLACE, <WP_TASKSINFO>
; #endif
;
; Remote access client folder
; #if LDR
; Create network folder, if it does not exist
WPFolder, <WP_NETWORK>, "Remote Access Client", UPDATE, <WP_CONNECTIONSFOLDER>
; Create a Network Services folder, if it does not exist
WPFolder, <WC_NETSERV>, "Network Services" , UPDATE, <WP_NETWORK>,
SHOWALLINTREEVIEW=YES
```

```

; Now the Remote Access icon is being moved directly to Network Services
; Create a Remote access Client folder in Network Services
;   WPFolder, <WC_REMOTEAACCESS>, "Remote Access Client", UPDATE, <WC_NETSERV>
; Create Remote Access Client Guide
;   WPProgram, <LD_CLIENT_GUIDE>, "Remote Access Client Guide", UPDATE, <WP_TASKSINFO>
; Create LAN Distance Advanced Guide
;   WPProgram, <LD_ADV_GUIDE>, "LAN Distance Advanced Guide", UPDATE, <WP_TASKSINFO>
; Create LAN Distance remove object
;   WPProgram, <LDCS_REMOVE>, "LAN Distance Remove", UPDATE, <WP_INSTREMFOLDER>,
EXENAME=LDREMOVE.EXE;ICONFILE=\IBMINST\ICONS\1033.ICO
; Create shadow of LAN Distance icon in system setup
;   WPSHadow, <WP_WALX_SHAD>, "LAN Distance", UPDATE, <WP_CONFIG>, SHADOWID=<WP_WALX>
#endif
;
;
;
;
; #if NW
; Create a Network Services folder, if it does not exist
;   WPFolder, <WC_NETSERV>, "Network Services", UPDATE, <WP_NETWORK>, SHOWALLINTREEVIEW=YES
; Create program - Delete NetWare Requestor
;   WPProgram, <CB_NetWare_REMOVE>, "Delete NetWare Requestor", UPDATE, <WP_INSTREMFOLDER>
; Create program - Delete Network Signon coordinator
;   WPProgram, <NSC_REMOVE>, "Delete Network Signon Coordinator", UPDATE,
<WP_INSTREMFOLDER>, EXENAME=\IBMINST\DELNSC.CMD
#endif
;
; #if OS2PEER
; Create a Network Services folder, if it does not exist
;   WPFolder, <WC_NETSERV>, "Network Services", UPDATE, <WP_NETWORK>, SHOWALLINTREEVIEW=YES
; shadow of readme
;   WPSHadow, <LS_README_PEER_SHAD>, "Peer Services Readme", UPDATE,
<WP_READMEFOLDER>, SHADOWID=<LS_README_PEER>
; Create program - Delete Network Signon coordinator
;   WPProgram, <NSC_REMOVE>, "Delete Network Signon Coordinator", UPDATE,
<WP_INSTREMFOLDER>, EXENAME=\IBMINST\DELNSC.CMD
#endif
;
; #if NETFIN
; NetFinity client guide
;   WPProgram, <NFCLIENTGUIDE>, "NetFinity Client Guide", UPDATE, <WP_TASKSINFO>,
EXENAME=VIEW.EXE;PARAMETERS=\OS2\BOOK\SERVICES.INF
; shadow of readme
;   WPSHadow, <NFPREADME_SHAD>, "NetFinity Client Readme", UPDATE,
<WP_READMEFOLDER>, SHADOWID=<NFPREADME>
#endif
;
[delete]
;
; #if SVAGENT
; delete DMI and DPI Programmers guides
;   <SVADPI>
;   <SVAPRG>
#endif
;
[move]
; VoiceType will be done here for Beta only:
; Move VoiceType folder to Programs
;   <WP_SPEECH>, <WP_PROGRAMSFOLDER>
; Move VoiceType Users Guide to Tasks
;   <SPCH_USERGUIDE>, <WP_TASKSINFO>
;
; Security Services into System Setup

```

```

        <UPM_FOLDER>, <WP_CONFIG>
;
; #if MPTS
;   MPTS icon into system setup (in case Peer created it)
;   <NTS_MPTS_ICON>, <WP_CONFIG>
; #endif
;
; #if OS2PEER
;   Move LS install/config to Install/Remove folder
;   <LS_INSTALL>, <WP_INSTREMFOLDER>
;   Move Peer user guide to Tasks
;   <PEERUser>, <WP_TASKSINFO>
;   Move Main Peer folder to Network Services
;   <LS_FOLDER>, <WC_NETSERV>
;   Move workstation to Network Services
;   <PEER_WKST>, <WC_NETSERV>
;   Move LS Admin GUI to Network Services
;   <LS_ADMIN>, <WC_NETSERV>
;   Move File and Print Audit log to Utilities
;   <LS_AUDIT>, <WP_TOOLS>
;   Move File and Print Error log to Utilities
;   <LS_ERROR>, <WP_TOOLS>
;   Move Network DDE & Clipboard to Utilities
;   <LS_CLIP>, <WP_TOOLS>
;   Move Network Messaging to Utilities
;   <LS_NETMSG>, <WP_TOOLS>
;   Move Password Coordination Guide to Tasks
;   <NSC_REF>, <WP_TASKSINFO>
;   Move Password Coordination Folder to Utilities
;   <NSC_FOLDER>, <WP_TOOLS>
; #endif
;
; #if NW
;   Move NetWare install to Install/Remove
;   <NVL_INSTALL>, <WP_INSTREMFOLDER>
;   Move NetWare Client Guide to Tasks
;   <NVL_CLIENT>, <WP_TASKSINFO>
;   Move Main Novell folder to Network Services
;   <NOVELL_FOLDER>, <WC_NETSERV>
;   Move Password Coordination Guide to Tasks
;   <NSC_REF>, <WP_TASKSINFO>
;   Move Password Coordination Folder to Utilities
;   <NSC_FOLDER>, <WP_TOOLS>
; #endif
;
;
; #if LDR
;   Move Remote Access program into Remote Access Folder
;   <WP_WALX>, <WC_NETSERV>
; #endif
;
; #if SVAGENT
;   Move systemview startup out of startup folder, into main SysView folder
;   <CASOS2>, <KARATCA>
;   Move SystemView Agent Remove to Install/Remove
;   <CAGINSTS>, <WP_INSTREMFOLDER>
;   Move SystemView Management Agent Guide to Tasks
;   <SVAUSR>, <WP_TASKSINFO>
;   Move SystemView Agent System Management Agent to Utilities
;   <KARATCA>, <WP_TOOLS>
; #endif
;
; #if NETFIN

```



```

; Move PC SystemView System Management Client to Utilities
<RPSM_FLD>, <WP_TOOLS>
#endif
;
;
; #if MFS
; Move MFS Status: to Mobile Office Services folder
<MFSOBJECT>, <MARSCML>
; Move Mobile office Services Guide to Tasks
<MCMDOC>, <WP_TASKSINFO>
; Move Mobile Office Services folder to Network Services
<MARSCML>, <WC_NETSERV>
#endif
;
;
[shadow]
;
;
[copy]
;
;
[update]
;
;
;
; #if MPTS
; rename MPTS to MPTS^Network Adapters^and Protocol Services
<NTS_MPTS_ICON> TITLE=MPTS^Network Adapters^and Protocol Services
; assign exe file for DHCP Monitor
SETUPSTRING <WS_DHCP_MONITOR> EXENAME=DHCPMON.EXE
; assign exe file for DDNS Configuration
SETUPSTRING <MPTS_DDNS_OBJECT_SETUP> EXENAME=DDNSCFG.EXE
; Set exe and inf file name for NAPS Guide
SETUPSTRING <MPTSCfg> EXENAME=VIEW.EXE
SETUPSTRING <MPTSCfg> PARAMETERS=\OS2\BOOK\MPTSCFG.INF
; Set exe and inf file name for DHCP Admin guide
SETUPSTRING <NP_DHCP_GUIDE> EXENAME=VIEW.EXE
SETUPSTRING <NP_DHCP_GUIDE> PARAMETERS=\OS2\BOOK\DHCPCLNT.INF
; Set exe and inf file name for Dynamic IP Intro
SETUPSTRING <NP_DIP_INTRO> EXENAME=VIEW.EXE
SETUPSTRING <NP_DIP_INTRO> PARAMETERS=\OS2\BOOK\DIPINTR.INF
#endif
;
; #if NETFIN
; Rename NetFinity to NetFinity System Management Client
<RPSM_FLD> TITLE=NetFinity^System Management Client
; TME 10 NetFinity^System Management Client^Readme
<NFPREADME_SHAD> TITLE=TME 10 NetFinity^System Management Client^Readme
#endif
;
; #if SVAGENT
; Rename SystemView folder
<KARATCA> TITLE=SystemView Agent^System Management Agent
; Rename Deinstall utility to SystemView Agent Remove
<CAGINSTS> TITLE=SystemView Agent Remove
; Rename SystemView Management Agent Guide
<SVAUSR> TITLE=SystemView Management Agent Guide
#endif
;
; #if NW
; Rename Netware Services folder
<NOVELL_FOLDER> TITLE=Netware Services
; Assign cmd file for NetWare Remove

```

```

        SETUPSTRING <CB_NetWare_REMOVE> EXENAME=\IBMINST\DELNW.CMD
;   Rename NetWare Client Install
    <NVL_INSTALL> TITLE=NetWare Client Install
;   Rename NSC folder Password^Coordination
    <NSC_FOLDER> TITLE=Password^Coordination
;   Rename SignOn Password^Coordination
    <NSC_PM> TITLE=Password^Coordination
;   Rename Server Password Coordination^Server
    <NSC_SVR> TITLE=Password Coordination^Server
;   Rename Retry Password Coordination^Retry Process
    <NSC_RETRY> TITLE=Password Coordination^Retry Process
;   Rename Reference Password Coordination Guide
    <NSC_REF> TITLE=Password Coordination Guide
;   Assign cmd file for Password Coordination Remove
    SETUPSTRING <NSC_REMOVE> EXENAME=\IBMINST\DELNSC.CMD
#endif
;
;#if OS2PEER
;   rename LS install/config
    <LS_INSTALL> TITLE=File and Print Client^Install/Remove
;   rename OS/2 Peer to Logons
    <LS_FOLDER> TITLE=Logons
;   rename Start OS/2 Peer to Start^File and Print Client
    <LS_START> TITLE=Start^File and Print Client
;   rename Peer Workstation Logon to File and Print Client^Workstation Logon
    <PEER_LOCALLOGON> TITLE=Print Client^Workstation Logon
;   readme Network^Readme
    <LS_README_PEER_SHAD> TITLE=Network^Readme
;
;   Rename NSC folder Password^Coordination
    <NSC_FOLDER> TITLE=Password^Coordination
;   Rename SignOn Password^Coordination
    <NSC_PM> TITLE=Password^Coordination
;   Rename Server Password Coordination^Server
    <NSC_SVR> TITLE=Password Coordination^Server
;   Rename Retry Password Coordination^Retry Process
    <NSC_RETRY> TITLE=Password Coordination^Retry Process
;   Rename Reference Password Coordination Guide
    <NSC_REF> TITLE=Password Coordination Guide
;   Assign cmd file for Password Coordination Remove
    SETUPSTRING <NSC_REMOVE> EXENAME=\IBMINST\DELNSC.CMD
#endif
;
;#if MFS
;   rename Mobile Office Services
    <MARSCM1> TITLE=Mobile Office Services
;   rename Start MFS
    <MCMPROG> TITLE=Start^Mobile Office Services
;   rename Stop MFS
    <MCMSTOP> TITLE=Stop^Mobile Office Services
;   rename MFS doc
    <MCMDOC> TITLE=Mobile Office Services Guide
#endif
;
;#if LDR
;   Rename Remote Access Client
    <WP_WALX> TITLE=Remote Access Client
;   Set exe and inf file name for Remote Access Client Guide
    SETUPSTRING <LD_CLIENT_GUIDE> EXENAME=VIEW.EXE
    SETUPSTRING <LD_CLIENT_GUIDE> PARAMETERS=\OS2\BOOK\A9M04MST.INF
;   Set exe and inf file name for Remote Access Advanced Guide
    SETUPSTRING <LD_ADV_GUIDE> EXENAME=VIEW.EXE
    SETUPSTRING <LD_ADV_GUIDE> PARAMETERS=\OS2\BOOK\A3T12MST.INF

```

```
#endif  
;  
;
```

A.5.7 The SHUF_KEY.INI File

This file specifies the components that were installed and ready for being rearranged on the desktop.

```
;  
; The Keyword file the for ClnDesk.EXE to control execution  
; Keywords to use: OS2PEER TCPIP MPTS LDR NW NETFIN SVAGENT MFS  
;  
Products=OS2PEER TCPIP MPTS LDR NW NETFIN SVAGENT MFS  
;  
;
```

A.6 TME 10 SD 3.1.3 Base Configuration Parameter

The following table discusses TME 10 SD 3.1.3 configuration parameters.

Table 24. Base Configuration File Parameters (Part 1 of 4)

Keyword	Description
API TRACE FILE SIZE	The maximum size of the Software Distribution API trace file in bytes. When the file is full, it is automatically backed up and a new trace file is generated.
AUTHORIZE	This keyword authorizes pull mode clients to install a software object or to execute a data file. NONE: No target is authorized to install unless it is authorized through the <code>auth</code> command. ALL: Any target is authorized, unless it is unauthorized through the <code>unauth</code> command. Set this keyword as soon as you have installed the server.
AUTOMATIC TARGET INVENTORY	This causes clients to perform a automatic hardware and software inventory and store the result in the server database. YES: Inventory is performed when the client is added. NO: Default
AUTOMATIC TARGET REGISTRATION	Clients automatically register themselves in the server database as one of the server local targets. YES: Enables targets to be registered at the server. NO: Default To perform a automatic registration, the <code>TARGET_ADDRESS</code> and the <code>TARGET_MODE</code> keywords must be set in the clients base configuration file. The <code>CONFIGURATION</code> keyword must be <code>CLIENT</code> .
BACKUP AREA	Name of the directory where the software objects are stored when they are installed as removable.
CONFIGURATION	The configuration of the Software Distribution for OS/2 node. The name is set up during the installation and cannot be changed. <code>SERVER_NO_COMMS</code> : Server with no remote connections. <code>SERVER_WITH_COMMS</code> : Server with connections to other servers.
CONNECTION WINDOW DURATION	This value becomes the default for the <code>d</code> operation on the connect command. It represents the amount of time in minutes that the connection window from the server to the client is to remain open.
DACA IDLE TIME	The time in seconds after which an idle client/server connection is considered as failed.

Table 25. Base Configuration File Parameters (Part 2 of 4)

Keyword	Description
DACA RETRY TIME	The time in seconds before a failed client/server connection is retried.
EXTEND FILE SYSTEM ON STORE	This keyword specifies whether the product is enabled to extend the file system on the destination workstation if there is not enough space to store the file during a send or retrieve operation. YES: If there is not enough space the destination file system is extended. NO: The file system cannot be extended.
FILESYSTEM	An indicator of the files systems that the operating system supports. 1 - UNIX based 2 - OS/2, Windows, DOS 3 - NetWare 5 - Windows NT
INVENTORY_PROGRAM	Name of the program invoked when the <code>inv</code> command is issued.
LAN AUTHORIZATION	0 - LAN address authorization is not required. 1 - LAN address validation. If you specify 1, you must supply the LAN address of each local target.
LOG FILE SIZE	The maximum size if the message log file in bytes.
MACHINE TYPE	The operating system in use on your Software Distribution OS/2 node. The name is set up during installation and cannot be changed.
MAX CONNECTION	The maximum number of local targets allowed to process requests simultaneously. max = 50
MAX LOCAL TARGETS	The maximum number of local targets that can be configured for a server, max = 1000
MAX REQUESTS	The maximum number of requests that can be active simultaneously. 1 - 65536
MAX SERVER CONNS	The maximum number of connections to remote server DACAs that can be open simultaneously. 0-256

Table 26. Base Configuration File Parameters (Part 3 of 4)

Keyword	Description
MAX SERVER TARGETS	The total number of non-adjacent server targets that can be configured for a server. max = 2048
MAX SNADS ROUTER	The maximum number of SNADS connections that can be defined. max = 800
MAX STS	The maximum number of STS processes that can be simultaneously handling connections to remote distribution servers. 0-256
MAX STSROUTES	The maximum number of STS connection that can be defined. max = (1600 - number of SNADS routes defined)
MAX TRAGETS	The total number of local and/or remote targets that can be configured. Must be greater than the sum of MAX LOCAL TARGETS and MAX SERVER TARGETS. max = 40000
MAX USER INTERFACES	The maximum number of simultaneous GUI and command line sessions.
MESSAGE LOG LEVEL	This file defines the log level that should be used by distribution clients before they establish a connection to the distribution server and discover the level configured for them there. M - minimal N - normal D - diagnostic
PROTOCOL	A transmission protocol used to communicate with other servers and clients. <protocol name> <address> protocol name - TCP, IPX, NBI (NetBIOS), SNA
REPOSITORY	The name of the repository where the objects are stored. Default is \SOFTDIST\REPOS.

Table 27. Base Configuration File Parameters(Part 4 of 4)

Keyword	Description
Server	The system name of the distribution server. If you want to attach a target to other distribution servers, you must name those servers here. Up to five server names can be specified. The first name defines the distribution server; the other entries define servers that can be administered from this client.
SERVICE AREA	The name of the directory for storing files and information about software objects that require activation. Default is \SOFTDIST\SERVICE.
STS IDLE TIME	The time in seconds after which an idle STS connection is considered as failed. 0 - 32767
STS RETRY TIME	The time in seconds before a failed STS connection is retried. 0 - 32767
TARGET PASSWORD AUTHENTICATION	Activates target password authentication. YES NO - default
TRACE FILE SIZE	The maximum size of internal trace files in bytes.
WORK AREA	The name of the directory where temporary work files are created and used. Default \SOFTDIST\WORK.
WORKSTATION NAME	The name of the workstation that is running your Software Distribution for OS/2 node. The name is set up during installation and cannot be changed.

A.6.1 Sample Base Configuration File

```
# BASE CONFIGURATION FILE
#
# This file should be stored as d:\SOFTDIST\nvdm.cfg
WORKSTARTION NAME:      SDSERV1
SERVER:                  SDSERV1
PROTOCOL:                NBI 400052003170 0 20
REPOSITORY:              C:\SOFTDIST\REPOS
WORK AREA:                C:\SOFTDIST\WORK
BACKUP AREA:             C:\SOFTDIST\BACKUP
SERVICE AREA:           C:\SOFTDIST\SERVICE
CONFIGURATION:           SERVER_NO_COMMS
MESSAGE LOG LEVEL:      D
LOG FILE SIZE:           200000
API TRAVE FILE SIZE:    200000
TRACE FILE SIZE:        200000
MAX USER INTERFACES:    20
MAX ATTEMPTS:           5
MAX TARGET:              600
TARGET PASSWORD AUTHENTICATION: NO
AUTHORIZATION:           NONE
AUTOMATIC CLIENT UPGRADE: NO
LAN AUTHORIZATION:       0
AUTOMATIC TARGET REGISTRATION: YES
DACA RETRY TIME:        300
MACHINE TYPE:           OS/2
INVENTORY PROGRAM:      C:\SOFTDIST\BIN\FNDINV.EXE
```


Appendix B. More Hints and Tips for OS/2 Warp 4 Installers

This appendix contains helpful information for people who install OS/2 Warp 4 systems and who help users migrate to OS/2 Warp 4 systems.

B.1 Using RSPDDI for Device Driver Installation

This section describes using `RSPDDI` with CID for device driver installation. Use the `RSPDDI` command to write the required `ABIOS.DDP` file.

RSPDDI Syntax

```
RSPDDI x:\ddp c:\[os2] x:\ddp\abios.ddp
```

The `RSPDDI` utility takes the following parameters. All are position dependent, and all are required.

<code>x:\ddp</code>	The source location for the DDI files.
<code>c:\os2</code>	The destination location where the device drivers go. If the destination is <code>c:\</code> instead of <code>c:\os2</code> , the files are placed in the directory specified by the <code>.ddp</code> file.
<code>x:\ddp\abios.ddp</code>	The name of the DDP file to use.

Example:

```
:TITLE
Title of Device Driver
* REM updates to the CONFIG.SYS below
:CONFIG
BASEDEV=IBMRAID.ADD
* REM files to copy over
:FILES
IBMRAID.ADD \OS2\BOOT\IBMRAID.ADD
* REM end of .DDP file
* REM updates to OS2.INI
:OS2INI
app_name key_name value_data
```

See the *OS/2 Command Reference* for more information about the command.

B.2 Installing OS/2 Printer FixPaks Using RINSTPRN

This section tells how to use the Remote Printer Install utility to install new OS/2 Printer FixPaks. Only the installation of the OS2 print drivers was tested, not the WIN-OS2 print drivers. The example below used the LASERJET driver FixPak.

1. Create a printer FixPak directory on the CID server. This page refers to that directory as FIXP12.
2. Create a directory, PMDD_1, under FIXP12.
3. XCOPY the printer FixPak diskette into FIXP12\PMDD_1.
4. Copy \OS2\INSTALL\PRDESC.LST into FIXP12.
5. Copy \OS2\INSTALL\PRDRV.LST into FIXP12.
6. Copy \OS2\MDOS\WINOS2\SYSTEM\CONTROL.INF into FIXP12.
7. Copy \OS2\MDOS\WINOS2\SYSTEM\DRVMAP.INF into FIXP12.
8. Edit and modify PRDESC.LST so that only printers that use the print driver in the FixPak are in the list. For example:

```
Brother HL-660: Brother HL-660 (LASERJET.DRV)
Brother HL-6V: Brother HL-6V (LASERJET.DRV)
Epson ActionLaser 1000/1500: Epson ActionLaser 1000/1500 (LASERJET.DRV)
Epson ActionLaser 1600: Epson ActionLaser 1600 (LASERJET.DRV)
Epson ActionLaser II: Epson ActionLaser II (LASERJET.DRV)
Epson EPL-7000: Epson EPL-7000 (LASERJET.DRV)
Epson EPL-8000: Epson EPL-8000 (LASERJET.DRV)
HP Color LaserJet: HP Color LaserJet (LASERJET.DRV)
HP DeskJet 1200C: HP DeskJet 1200C (LASERJET.DRV)
HP DeskJet 1600C: HP DeskJet 1600C (LASERJET.DRV)
HP LaserJet 2000: HP LaserJet 2000 (LASERJET.DRV)
HP LaserJet 4: HP LaserJet 4 (LASERJET.DRV)
HP LaserJet 4 Plus: HP LaserJet 4 Plus (LASERJET.DRV)
HP LaserJet 4L: HP LaserJet 4L (LASERJET.DRV)
HP LaserJet 4M: HP LaserJet 4M (LASERJET.DRV)
```

9. Edit and modify PRDRV.LST so that the only driver listed matches the driver name for the FixPak driver. For example:

```
IBMNULL.DR_    1      IBM Null Printer Driver
LASERJET.DR_   1      HP Laserjet III Printer Driver
PSCRIPT.DR_    1      PostScript Printer Driver
HPDJPM.DR_     1      HP Deskjet Printer Driver
IBM52XX.DR_    1      IBM 52XX Printer Driver
EPSON.DR_      1      Epson Printer Driver
PLOTTERS.DR_   1      Plotter Driver
```

```

PMPLTPD.DR_ 1      PMPLOT Queue Processor Driver
OMNI.DR_    2      Omni driver
IBM4019.DR_ 2      IBM 4019 Printer Driver
IBM42XX.DR_ 2      IBM 42XX Printer Driver
IBMPCL3.DR_ 2      IBM PCL3 Printer Driver
IBM52012.DR_ 4     IBM 5201-2 Printer Driver
SMGXPJET.DR_ 4     Paintjet Printer Driver
IBMPCL5.DR_ 5      IBM PCL5 Printer Driver

```

should be changed to:

```
LASERJET.DRV 1      HP Laserjet III Printer Driver
```

The 1 in the line above refers to PMDD_1. RINSTPRN looks for the PMDD_1 directory under FIXP12 if the /S: parameter is /S:X:\FIXP12. Set this to 1 if the print driver disk it is not already set to 1.

10. Execute RINSTPRN with /DSC: pointing to the new modified PRDESC.LST file. Create a response file for new FixPak print driver.

If you want to use the same print queue names that are currently in use, you need to know this information or retrieve it from the OS2*.INI files. You can use an INI editor for this, or you can use the following REXX program to list all INI appnames and keynames. The appname PM_SPOOLER_QUEUE will have keynames of print queues.

```

/* REXX to type all OS2*.INI info to the screen */
Call RxFuncAdd sysloadfuncs, rexxutil, sysloadfuncs
call sysloadfuncs
result2 = call SysIni('BOTH', 'ALL:', 'Apps')
if result \= 'ERROR:' then
  do i=1 to Apps.0
    result3 = call SysIni('BOTH', Apps.i, 'ALL:', 'Keys')
    if result \= 'ERROR:' then
      do j=1 to Keys.0
        Say Apps.i ' key: ' Keys.j
      end
    end
  end

```

11. Use RINSTPRN to perform a normal print driver install with the /S: source parameter pointing to the FIXP12 directory and with /DRV: and /DSC: pointing to the modified versions of PRDRV.LST and PRDESC.LST.

With some print driver packages, you must zip the files with the OS/2 utility, PACK2.EXE, that comes with the OS/2 Visual Age C++ package from IBM.

To install WIN-OS2 print drivers, you need to modify the additional files DRVMAP.INF and CONTROL.INF.

When updating the file, be sure match uppercase and lowercase letters.

With OS/2 Warp 4, you need to use the updated version of `RINSTPRN` that comes on the CD-ROM with this redbook. The problem is that the print drivers now come from either the OS/2 print driver diskettes or from the OS/2 Windows diskettes. Previous versions of `RINSTPRN` did not support the devided printer driver diskettes mechanism.

B.3 BIOS Problems on RAM-Loadable BIOS Systems

The current RAM-loadable BIOS systems are: PS/2s 9556/9557; 9585; Thinkpad 700/720 and the 9553. IBM PC700/300 series systems that use the PCI/MCA bus may be RAM-loadable BIOS systems. The newer PS/2 9577/9576 systems are also RAM-loadable BIOS.

To install OS/2 on these systems, copy the `ABIOS.SYS` and the `*.BIO` file from the system reference disk for that specific machine to the OS/2 Install disk.

If problems continue on the system, boot from disks to a command prompt and look on the system hard drive in the root directory, the `\OS2` directory and the `\OS2\BOOT` directory for any copy of `ABIOS.SYS` that may exist and replace it with the `ABIOS.SYS` from the system reference disk. Add the `*.BIO` file to the directory where `ABIOS.SYS` is.

There are several ways to get around the BIOS problems of the RAM-loadable BIOS machines. First, you must make or get a reference diskette for the machine. On the reference disk, you should find an `ABIOS.SYS` file, a `ABIOS.DDP` file, and a `*.BIO` file.

The official way is to use the DDI response file keywords. You will find the needed files for DDI keywords on the reference disk. You should create a separate directory on the server to put the `ABIOS.DDP`, `*.BIO`, and `ABIOS.SYS` file from the reference disk in. This procedure requires that you use a custom response file for these machines. This response file will not work on other machines. See the `SAMPLE.RSP` file for information about DDI keywords.

At times you will find it necessary to modify the CID boot diskettes by editing the `ABIOS.SYS` file to include the new `*.BIO` file and add the new `*.BIO` file to the diskette.

Another way to fix an BIOS problem is to modify the `ABIOS.SYS` file on the the CID server disk images. Find the name of the `*.BIO` file on the system

reference disk and put that name at the top of the ABIOS.SYS file on the CID server OS/2 disk images. The ABIOS files are on the Install disk, which is known as DISK_0 on the CID server. You must also put the *.BIO file on the disk with the ABIOS.SYS file and the CID server. Note that when you edit the ABIOS.SYS file and to add the new .BIO filename to the list, it must be in uppercase letters.

B.4 FDISK Install Notes

When using `FDISK`, we recommend setting the prompt to display return codes from executables. At an OS/2 command prompt, enter the following command:

```
SET PROMPT=[ $R $P ]
```

The command prompt return code (`$R`) from the `FDISK /QUERY` should be 0. If it is not, make note of what it is.

B.4.1 IDE Hard Drive BIOS Setup

It is the responsibility of the PC system hardware BIOS to provide access to the hard drive so that software programs can use the hard drive. This is done with an industry standard BIOS call into the system interrupt 13 (INT13). A problem with the industry standard is that it only allowed for 10 bits to define a cylinder to select for an operation. This limits the BIOS INT13 to only using 1024 cylinders or less. In the past, the BIOS INT13 code would map INT13 cylinder requests directly to physical cylinders on the IDE hard drive.

Because new IDE-type hard drives commonly have more than 1024 cylinders, the industry was confronted with a problem. To get around this problem, the industry has developed a new standard for INT13 BIOS with the use of a LBA (Logical Block Addressing) mode for IDE-type hard drives. The new LBA mode BIOS INT13 code will take the drive head count and increase it logically while it decreases the cylinder count.

Normal conversions will double the head count and cut the cylinder count in half. They will also quadruple the head count and cut the cylinder count to one-fourth. Some new BIOS's allow for eight times the head count and cut the cylinder count to one-eighth.

When you configure a new hard drive for a system, it is normally best to make sure that the INT13 BIOS can map all hard drive cylinders into the 1024 range.

B.4.2 FDISK Return Code Information

Returns from the parser are two bytes in the following forms:

- High order byte: rstatus values

Possible values contained in high order byte:

Dec	Goal
10	Get num of fixed disks
12	Num of fixed Disks = 0
14	Set total num handles
16	Get handles for disks
18	Disk control
20	Read from the MBR
22	Write to the MBR
24	Read from MBB sector
26	Write to MBB sector
28	No MBB partition
30	Bad signature/no mbbc
32	Read from the MBB vol
34	Write to the MBB vol
36	Read MBB alias Table
38	Write MBB alias Table
40	Write bootsect
42	Write to the MBB vol
44	DiskParm[0] = NIL
46	Create new IBMRec
48	Disk class empty
50	Diskio of BootRec Fail
52	Diskio of BootRec Fail
54	Read MBR Fails
56	Write to MBR Fails

- Low order byte: Last return code

Some possible return codes:

Dec	Interpretation
0	No error
1	Install partition not formatted
2	Install partition too small
3	Disk not partitioned
4	Reboot Required
7	Fatal FDISK Error
8	Simple Install

B.4.3 FDISK Command Line Examples

The following is a list of some sample `FDISK` command lines:

- To create a bootmanager partition using the first available free space at the start of the drive:

```
FDISK /CREATE /BOOTMGR /START:T
```

- To create a primary partition at the top of first drive and add this partition to the bootmanager as OS2V40:

```
FDISK /CREATE:OS2V40 /START:T /DISK:1 /VTYPE:1 /SIZE:300
```

- To create a logical partition on disk 1 using all free disk space:

```
FDISK /CREATE /DISK:1 /VTYPE:2
```

- To delete all partitions on second disk. This will not delete any SYSTEM (IML/FE) type partitions.

```
FDISK /DELETE:ALL /DISK:2
```

- To delete the logical FAT partition on first disk:

```
FDISK /DELETE /VTYPE:2 /FSTYPE:FAT /DISK:1
FDISK /DELETE /VTYPE:2 /FSTYPE:06 /DISK:1
```

- To delete the 400 MB partition on disk 2:

```
FDISK /DELETE /SIZE:400 /DISK:2
```

- To delete a bootable logical partition:

```
FDISK /DELETE /DISK:1 /BOOTABLE:1 /VTYPE:2
```

- To delete the partition on disk 1 (01) with volume name 00101800:

```
FDISK /DELETE /VOLID:0100101800
```

- To delete the partition with the bootmanager name WARP:

```
FDISK /DELETE /NAME:WARP
```

- To pass a response file to FDISK for processing:

```
FDISK /FILE:FDISK.RSP
```

- FDISK response file example

The following is a sample FDISK response file:

```
/CREATE,/BOOTMGR,/DISK:1
/CREATE:OS2V4,/SIZE:300,/VTYPE:1,/DISK:1
/CREATE:MAINT,/SIZE:20,/VTYPE:2,/DISK:1,/START:B
/CREATE,/VTYPE:2,/DISK:1
/CREATE,/VTYPE:1,/DISK:2
```

There is a restriction on the size of a FDISK command line. If a command line that is too long, the last parameter or so will not be used.

B.5 OS/2 Warp 4 CD-ROM Installation Notes

OS/2 Warp 4 has 3 boot diskettes:

- | | |
|------------|--|
| Diskette 0 | The first diskette, the install diskette, contains the kernel and the BIOS files, *.BIO. You should not need to make any changes to this diskette unless your system is a RAM-loadable BIOS system. Almost all RAM-loadable BIOS systems are newer IBM PS/2 systems. |
| Diskette 1 | The second diskette contains the CONFIG.SYS, SNOOP.LST, and BASEDEV device drivers. Most changes to this diskette involve editing the CONFIG.SYS and SNOOP.LST. |
| Diskette 2 | The third diskette (contains the DEVICE device drivers, IFS (Installable File System) drivers, .DLL files, CMD.EXE, and so forth. Usually, you do not change this diskette. |

The following are base changes to the CONFIG.SYS that are a starting point to get the CDROM recognized. If you have a problem, a starting point is to remove all drivers that are not specifically needed for the problem system.

```
buffers=32
iopl=yes
memman=swap,delayswap
protshell=sysinst1.exe
REM set os2_shell=sysinst2.exe          not used!
diskcache=D2,LW
```



```

protectonly=yes
libpath=.;\os2\dll;\os2\install;
ifs=hpfs.ifs /c:64
pauseonerror=no
codepage=850
devinfo=kbd,us,keyboard.dcp
devinfo=scr,ega,vtbl850.dcp
device=\dos.sys
REM device=\mouse.sys          this driver is never used for
REM                               first phase of install!
set path=\;\os2;\os2\system;\os2\install
set dpath=\;\os2;\os2\system;\os2\install
set keys=on
basedev=ibmkbd.sys
basedev=ibm1flpy.add
basedev=ibm1s506.add
REM basedev=ibm2flpy.add      only needed for PS/2 MCA bus systems!
REM basedev=ibm2adsk.add     only needed for PS/2 MCA bus systems!
REM basedev=ibm2scsi.add     only needed for PS/2 MCA bus systems!
basedev=ibmint13.i13
basedev=os2dasd.dmd
device=\testcfg.sys
REM basedev=xdfloppy.flt     not needed for CDROM install!
REM set os2_shell=cdinst.exe not used!
set saveconnect=1
set cdrominst=1
ifs=cdfs.ifs /q
REM basedev=ahal52x.add
REM basedev=ahal54x.add
REM basedev=ahal64x.add
REM basedev=ahal74x.add
REM basedev=aic7770.add
REM basedev=aic7870.add
REM basedev=btscsi.add
REM basedev=fd16-700.add
REM basedev=fd8xx.add
REM basedev=fd7000ex.add
REM basedev=dpt20xx.add
REM basedev=dac960.add
REM basedev=flashpt.add
REM basedev=ipsraid.add
REM basedev=ql10os2.add
REM basedev=ql40os2.add
REM basedev=ql510.add
REM basedev=chincds1.flt
REM basedev=hitcds1.flt
REM basedev=neccds1.flt

```

```
REM basedev=sonycds1.flr
REM basedev=toshcde1.flr
REM basedev=ibmidecd.flr
REM basedev=tmv1scsi.add
REM basedev=sony535.add
REM basedev=lms206.add
rem 154569 basedev=lms205.add
REM basedev=mitfx001.add
REM basedev=sbcd2.add
REM basedev=sony31a.add
device=\os2cdrom.dmd
set os2_shell=cdboot.exe
set oemprogram=\ibminst\npconfig.exe
set exitwhendone=1
```

The next step is to add the specific CD-ROM drivers needed for the problem PC system. The most commonly used CD-ROM driver at this time is IBMIDECD.FLT, which works with IBM1S506.ADD driver for CD-ROM driver connected to IDE controllers. For information about parameter switches for IBM1S506 and information about device drivers, enter the following command from an OS/2 command prompt:

```
HELP BASEDEV
```

The second most commonly used driver is the SBCD2.ADD, used for the CD-ROM driver connected to a SoundBlaster adapter.

If the CD-ROM drive is not either IBMIDECD.FLT or SBCD2.ADD, then you must provide information about the CD-ROM type and connection adapter. Most other type of CD-ROMs are connected to SCSI adapters; so you will need to figure out what SCSI driver from above to use. After you select the correct SCSI driver, you need to select the correct CD-ROM .ADD driver or .FLT driver to work with the SCSI driver.

B.5.1 Updating SNOOP.LST and .SNP Files

OS/2 Warp 4 now has a SNOOP.LST file and .SNP files. In many cases, you must edit the SNOOP.LST file on DISK1 and remove or remark out .SNP files from the lists. You need to remove or remark out all .SNP file where you have removed or remarked out the corresponding CONFIG.SYS driver.

You can also remove the PCI.SNP if you do not have a PCI bus system and PCMCIA.SNP if you do not have PCMCIA hardware.

You can remove all audio snoopers if you do not have a sound card.

The IR.SNP is for InfraRed driver support which is mostly found on laptop systems.

The following is the standard SNOOP.LST:

```
resrv.snp
netdet1.snp
ibkbd.snp
ibm1flpy.snp
ibm1s506.snp
; SCSI Snoopers
aha6360.snp
aha154x.snp
aha174x.snp
aic7870.snp
ql10os2.snp
ql40os2.snp
ql510.snp
ipsraid.snp
btscsi.snp
fd16-700.snp
fd8xx.snp
fd7000ex.snp
dpt20xx.snp
flashpt.snp
dac960.snp
; Misc. Snoopers
pcmcia.snp
ir.snp
netdet2.snp
; CDROM Snoopers
mitfx001.snp
sbcd2.snp
sony31a.snp
; Audio Snoopers
sndgalax.snp
auddrive.snp
jazz16.snp
sndblast.snp
pas16.snp
bsaudio.snp
; Misc. Snoopers
parallel.snp
mouse.snp
serial.snp
; Note: Place additional snoopers above this line
pcibus.snp
```

OS/2 Warp 4 has a Device Driver Pak CD-ROM. You can use a Web browser to view this CD-ROM. This will point to FTP sites and extra device drivers for OS/2 Warp 4.

B.5.2 Adding and Changing Device Drivers

Do not add `SET COPYFROMFLOPPY=1` to the `CONFIG.SYS` unless you have added a new driver to diskette 1 or diskette 2 to replace an existing driver on the disk. If you are adding a driver that was not present before, you do not need any modifications to the `CONFIG.SYS`. The `SAVECONNECT=1` works for new drivers.

Incorrect use of this statement can cause problems. In addition, if you add a new `BASEDEV` driver, you must copy the file to both Diskette 1 and Diskette 2.

B.5.3 System Hang During Bootup

If the system hangs during bootup and the user never gets an error, first try using the `ALT-F2` bootup function. Press **ALT-F2** when the little square with OS/2 appears in the upper-left corner. Watch to see what the last driver to execute is before the hang. If possible, remove this driver from the `CONFIG.SYS`. Some drivers must always be present, for example, `OS2DASD.DMD`.

B.5.4 Using `RESERVE.SYS` to Solve a Problem

For some install problems, you need to use the `RESERVE.SYS` driver, which can block other drivers from touching specific IO ports, and so forth, that may cause system hangs. For example, a customer has 100 systems of the same type. OS/2 installs without any problem on all systems that do not have a specific type adapter.

You need to get the hardware specification document for the specific adapter. Then add the `RESERVE.SYS` driver to the `CONFIG.SYS` with parameters to block the IO ports and so forth that the adapter uses.

B.6 Creating OS/2 Warp 4 NFS CID Boot Diskettes

This section tells how to create OS/2 Warp 4.0 NFS CID boot diskettes from the OS/2 Warp 4.0 CD-ROM.

You need the following to create the diskettes:

- The NFS Kit for TCP/IP 2.0. This kit is not part of the OS/2 Warp 4.0 CD-ROM.
- The NFS CSD UN57064 is recommended. The latest NFS CSDs are available on the FTP server:

`service.boulder.ibm.com`

The user ID is `anonymous`. The file is:

`/ps/products/tcpip/fixes/v2.0os2/un57064.`

Follow these steps:

1. Set up the CID server for NFS server capability. See the NFS documentation for information about setting up a NFS server.
2. Install the NFS Kit product image on the CID server. Place the diskette with NFS CSD UN57064 V2.0 into the A: drive and enter:

```
XCOPY A:\*.* <nfs_target_directory> /S /E /V
```

The NFS Kit product image is placed on the CID server in the `<nfs_target_directory>`.

3. Label three formatted, blank diskettes:

- Diskette 0
- Diskette 1
- Diskette 2

4. Insert Diskette 0 into drive A: and issue:

```
E:\CID\IMG\OS2WARP4\SEDISK.EXE /S:E:\OS2IMAGE /T:A:
```

E: is the drive containing either the OS/2 Warp 4.0 CD-ROM or the CID server system product image. Make sure SEDISK has been unpacked out of the CID macro file that resides on the seventh product diskette.

PCMCIA Support

If you are are creating boot diskettes for ThinkPads, add the `/P:` parameter to the `SEDISK` command and pass the line number in `PCMCIA.TBL` that describes the ThinkPad you want boot diskettes for. Note that `PCMCIA.TBL` is OS/2-version dependent and can be found in `\OS2\INSTALL` or in `\DISK_3\BUNDLE` in case of OS/2 Warp 4.

For example, to install OS/2 Warp 4 on a ThinkPad 760ED (Line 36 in `PCMCIA.TBL` from OS/2 Warp 4 represents this ThinkPad), issue:

```
D:\CID\IMG\OS2WARP4\SEDISK /S:D:\CID\IMG\OS2WARP4 /T:A: /P:36
```

5. When prompted, remove Diskette 0 and insert Diskette 1.
6. When prompted, remove Diskette 1 and insert Diskette 2.
7. If you are creating boot diskettes for a ThinkPad, see ThinkPad Steps for more information.
8. Modify the `CONFIG.SYS` file on Diskette 1. Inserting Diskette 1 into drive A: and make the following modifications to `CONFIG.SYS` to erase files that are not used during the installation process. This frees up space on the boot diskette for TCP/IP and NFS. Comment out the line for the mouse drive:

```
REM device=\mouse.sys
```

Change the `viotbl` line from:

```
devinfo=scr,vga,viotbl.dcp
```

to:

```
devinfo=scr,ega,VTBL850.dcp
```

9. Erase unnecessary files from Diskette 2. Insert Diskette 2 into drive A: and issue the following commands:

```
erase a:\unpack.exe  
erase a:\unpack2.exe  
erase a:\mouse.sys  
erase a:\viotbl.dcp  
erase a:\remview.exe  
erase a:\rminfo.dll
```

10. Keep Diskette 2 in drive A: and issue the following single-line command. Issue the command on one line; it is broken up here for ease of reading.

```
E:\CID\IMG\MPTS\THINLAPS.EXE
E:\CID\IMG\MPTS
A: IBMTOK.NIF /TCPIP
/IP:<target_machine_ipaddress>
/NM:<target_machine_netmask>
/H:<host_name>
/DMN:<DNS_domain_name>
/NSIP:<DNS_domain_nameserver_ipaddress>
/RT:<tcpip_default_router_ipaddress>
```

Adjust adapter information to your needs.

11. THINLAPS.EXE will indicate that it cannot update the A:\CONFIG.SYS file. This is to be expected.
12. Remove Diskette 2 and replace it with Diskette 1. Press **Enter** to complete the THINLAPS utility.
13. Remove Diskette 1.
14. Insert Diskette 2 into drive A: and issue the following commands:

```
erase a:\so32dll.dll
erase a:\tcp32dll.dll
```

SO32DLL.DLL and TCP32DLL.DLL are not required on the boot diskettes for NFS support. If other TCP/IP applications require these DLLs, you can place them on the CID server and specify their directory in the LIBPATH statement.

15. Issue the following command:

```
copy e:\cid\img\mptscsd\aflean.sys a:\afinet.sys
```

AFLEAN.SYS is a smaller version of AFINET.SYS only for CID Boot diskettes only. This version of AFINET.SYS saves approximately 100 KB on the CID boot diskette. Request this lean version of AFINET.SYS as APAR IP20945 or have the latest MPTS version handy, such as WR08415 or higher.

16. Use PKUNZIP2 to unzip THINNFS.EXE and NIC.MSG on the OS/2 Warp 4.0 CD-ROM in the E:\CID\IMG\MPTS\UTILITY\APPLETS\MPTSAPLT.ZIP file:

```
PKUNZIP2 MPTSAPLT.ZIP APPLETT/THINNFS.EXE APPLETT/NIC.MSG  
<nfs_target_directory>
```

The files are placed in the NFS Product Image subdirectory on the CID server. PKUNZIP2.EXE is in the E:\CID\IMG\MPTS subdirectory.

17. Before you install NFS support on the CID boot diskettes, copy the CONFIG.SYS file from Diskette 1 to Diskette 2.
18. Issue the following single-line command. Issue the command on one line; it is broken up here for ease of reading.

```
THINNFS  
  /D:<redirected_drive_letter>  
  /SRV:<server:path>  
  /T:A:  
  /TU:A:  
  /S:<nfs_product_image_directory>  
  /L1:<remote_log_file_name>  
  /HOST:<client_host_name>
```

Fill in the parameters as follows:

/D	The drive letter for the redirected drive
/SRV	The server and path as defined by the NFS server
/T	The target drive
/TU	The location of the CONFIG.SYS to be updated
/S	The source directory of the NFS product image
/L1	The drive and path of the remote log file name
/HOST	The client host name

19. Install LCU support, if necessary, by issuing the following single-line command:

```
E:\CID\LOCINSTL\CASINSTL.EXE  
  /TU:A:  
  /PA:<path_to_update_path_statement>  
  /CMD:<drive_and_path_of_lcu_command_file>  
  /PL:<paths_to_update_libpath_statement>  
  /REQ:<requester_name>
```

This step may not be necessary if you are not using the LCU CID process.

20. Copy the CONFIG.SYS file from Diskette 2 back to Diskette 1.

21. Now boot up with the NFS-enabled CID diskettes for OS/2 Warp 4. During boot-up, you can ignore the following error message appears:

```
can't find so32dll.dll
```

To remove this message, edit the MPTSTART.COM file on Diskette 2 and remove the `INETWAIT` statement from it.

B.6.1 Other MPTS CID Considerations When Using NFS

The MPTS response file should have the necessary sections to install TCP/IP support. You may use the following sample MPTS response as a template:

Parameters that may need to be changed:

[NETBEUI_NIF] section:

```
- NCBS=  
- SESSIONS=  
- NAMES=
```

[IFCONFIG] section:

```
- Address=  
- Netmask=
```

[ROUTE] section:

```
- Router=
```

[RESOLV] section:

```
- DOMAIN=  
- NAME=
```

The MPTS sample response file is configured for an IBM-compatible token-ring adapter (IBMTOK.NIF). If you are not using this adapter in the target system, then replace the following section:

```
PROT_SECTION = (  
NIF = IBMTOK.NIF  
SECTION_NAME = IBMTOK_nif  
|DriverName=IBMTOK$  
MAXTRANSMITS=6  
RECVBUFS=2  
RECVBUFSIZE=256  
XMITBUFS=1
```

with a `PROT_SECTION` that contains the appropriate MAC adapter driver. See the *MPTS Configuration Guide* for information on creating MPTS response files for particular MAC adapter cards.

```

INST_SECTION = (
UPGRADE_LEVEL = SAME
INSTALL = PRODUCT
)
PROT_SECTION = (
NIF = LANDD.NIF
SECTION_NAME = LANDD_nif
DriverName=LANDD$
Bindings=IBMTOK_nif
ETHERAND_TYPE="I"
SYSTEM_KEY=0x0
OPEN_OPTIONS=0x2000
TRACE=0x0
LINKS=40
MAX_SAPS=16
MAX_G_SAPS=0
USERS=6
T1_TICK_G1=255
T1_TICK_G1=15
T2_TICK_G1=3
T1_TICK_G2=255
T1_TICK_G2=25
T2_TICK_G2=10
IPACKETS=250
UIPACKETS=100
MAXTRANSMITS=6
MINTRANSMITS=2
TCBS=64
GDTS=30
ELEMENTS=1400
NETFLAGS=0x0
)

PROT_SECTION = (
NIF = NETBEUI.NIF
SECTION_NAME = NETBEUI_nif
DriverName=NETBEUI$
Bindings=IBMTOK_nif
ETHERAND_TYPE="I"
USEADDRREV="YES"
OS2TRACEMASK=0x0
SESSIONS=254
NCBS=254

```

```

NAMES=21
SELECTORS=50
USEMAXDATAGRAM="NO"
ADAPTRATE=1000
WINDOWERRORS=0
MAXDATARC=4168
TI=30000
T1=1500
T2=200
MAXIN=1
MAXOUT=2
NETBIOS_TIMEOUT=500
NETBIOS_RETRIES=3
NAMECACHE=1000
RND_OPTION=1
PIGGYBACKACKS=1
DATAGRAM_PACKETS=12
PACKETS=302
LOOP_PACKETS=8
PIPELINE=5
MAX_TRANSMITS=6
MIN_TRANSMITS=2
DLC_RETRIES=10
FC_PRIORITY=5
NET_FLAGS=0x0
)

PROT_SECTION = (
NIF = TCPIP.NIF
SECTION_NAME = TCPIP_nif
Bindings=IBMTOK_nif
DriverName=TCPIP$
)

PROT_NETBIOS = (
SECTION_NAME = NETBIOS
DRIVERNAME = NETBIOS$
ADAPTER0 = NETBEUI$,0
)

PROT_SECTION = (
NIF = IBMTOK.NIF
SECTION_NAME = IBMTOK_nif
)

```

```

DriverName=IBMTOK$
MAXTRANSMITS=6
RECVBUFS=2
RECVBUFSIZE=256
XMITBUFS=1
)

MPTS = (
[CONTROL]
Local_IPC = Yes
INET_Access = YES
NETBIOS_Access = No
[IFCONFIG]
Interface      = 0
Address        = 9.3.1.213          <===== CHANGE =====
Brdcast       =
Dest           =
Enable         = UP
Netmask        = 255.255.255.0     <===== CHANGE =====
Metric         = 0
Mtu            = 1500
Trailers       = NO
Arp            = NO
Bridge         = NO
Snap           = NO
Allrs          = NO
802.3          = NO
Icmpred        = NO
Canonical      = NO
EnableDhcp     = NO

[ROUTE]
Type           = default
Action         = add
Dest           =
Router         = 9.3.1.74          <===== CHANGE =====
Metric         = 1
)

* RESOLV SECTION
RESOLV = (
DOMAIN = itsorusi.itsc.austin.ibm.com <===== CHANGE =====
NAME = nameserver 9.3.1.74          <===== CHANGE =====

```

)

B.6.2 LCU Considerations

If you are using LCU, the LCU command file needs the `THINNFS` sections added to the command file. The following is an example of what needs to be added to the LCU command file for `THINNFS`:

```
x.nfs1 = 18
  x.18.name = 'NFS 1'
  x.18.statevar = 'CAS_' || x.18.name
  x.18.instprog = cidimgdir'\NFS\THINNFS.EXE ',
    'd:w ',
    'srv:palh32:d:/cid ',
    't:c: ',
    'tu:c:\ ',
    '/s:'cidimgdir'/nfs ',
    'host:palh33 ',
    'll:'logsdire'\nfs\nfs1.log'
  x.18.rspdir = ''
  x.18.default = ''

x.nfs2 = 19
  x.19.name = 'NFS 2'
  x.19.statevar = 'CAS_' || x.19.name
  x.19.instprog = cidimgdir'\NFS\THINNFS.EXE ',
    'd:z ',
    'srv:palh32:e:/ ',
    't:c: ',
    'tu:c:\ ',
    '/s:'cidimgdir'/nfs ',
    'host:palh33 ',
    'll:'logsdire'\nfs\nfs2.log'
  x.19.rspdir = ''
  x.19.default = ''
```

B.6.3 Known MPTS Problems and Workarounds

The following lists known MPTS problems that will be considered for future MPTS FixPaks.

- MPTS installed from CID boot diskettes fails to create the `RESOLV2` file in the `MPTN\ETC` subdirectory when the `resolv` section is in the MPTS response file. The problem is that the `RESOLV2` file is created on the boot diskettes instead of on the target drive.

- `THINLAPS` installs `SO32DLL.DLL` and `TCP32DLL.DLL` for NFS support. NFS does not require `SO32DLL.DLL` and `TCP32DLL.DLL`; thus a new option for `THINLAPS` will be `/NFS` and this will not copy these DLLs onto the target drive.
- `THINNFS` does not update the `LIBPATH` of the target `CONFIG.SYS`. The location of NFS's DLLs should be in the `LIBPATH` statement of the `CONFIG.SYS`.
- `THINNFS` does not prompt for the diskette which has the `CONFIG.SYS` file located on it.
- `MPTS WR08415` contains an updated, smaller "CID only version" of `AFINET.SYS` (`AFLEAN.SYS`) in order to provide more disk space on the CID boot diskettes.
- During boot-up with the NFS CID diskettes, the `can't find so32dll.dll` error message is displayed.
- If the `mount` statement in the `CONFIG.SYS` is executed with a `CALL` statement, for example:

```
CALL=C:\mount.exe -u0 -g0 w: cidserver:c:/cid
```

an error message may occur, `Press Enter to Continue...`, even when the mount is successful. You can work around this by putting `PAUSEONERROR=NO` in the `CONFIG.SYS` file.

Appendix C. Latest News on Netscape 2.02 Refresh and Java 1.1.4

This appendix informs you about how to remotely install the refresh version of Netscape 2.02 to support Java 1.1.4. Both software packages are available on the Internet. Point your Web browser to the following Web site:

<http://www.software.ibm.com/os/warp/swchoice>

IBM intranet users should point their Web browsers to the following Web site:

<http://antero.boulder.ibm.com/asd-bin/doc/>

In order to have Netscape 2.02 Refresh use Java 1.1.4, we recommend installing Java 1.1.4 prior to the refresh version of Netscape 2.02. The following sections detail the remote installation process:

- Installing Java 1.1.4 unattendedly
- Installing Netscape 2.02 Refresh unattendedly
- Installing Netscape 2.02 OS/2 Plug-ins unattendedly

C.1 Installing Java 1.1.4 Unattendedly

Unattended installations of Java 1.1.4 for OS/2 Warp are handled by `CLIFI`, the command-line interface to Feature Install, and can take 15 to 20 minutes or more. After the installation program finishes, the system must be restarted to complete the installation.

`CLIFI` requires two response files:

- The Java 1.1.4 for OS/2 Warp response file, for example, `JAVA114.RSP`
- The secondary response file, for example `CID.RSP`, where users can override the default selections.

The `CID.RSP` response file is illustrated later in this section. The primary response file, `JAVA114.RSP`, is too big to show here. It resides on the CD-ROM delivered with this book. As a prerequisite, Feature Installer 1.2.1 or higher is needed for Java 1.1.4 installations through CID. Feature Installer 1.2.1 is available from Software Choice, too.

C.1.1 Creating the Java 1.1.4 Response File

The `CID.RSP` file, as shown in Figure 149 on page 504, contains variables that allow you to select which components to install and the target drive and directory for each component, where appropriate.

```

java11.continue=UNATTENDED

runtime.selection=1
runtimeconfig.selection=1
java11.rundrv=D:
unicode.selection=1
unicodeconfig.selection=1

samples.selection=1
samplesconfig.selection=1
samples.smpdrv=D:
samples.smpath=\JAVA11

toolkit.selection=1
toolkitconfig.selection=1
toolkit.tktdrv=D:
toolkit.tktpath=\JAVA11

tlktdoc.selection=1
tlktdocconfig.selection=1
tlktdoc.tdocdrv=D:
tlktdoc.tdocpath=\JAVA11

debugger.selection=1
debuggerconfig.selection=1
debugger.dbgdrv=D:
debugger.dbgpath=\JAVA11\ICATJAVA

unifont.selection=1
unifontconfig.selection=1

ttengine.selection=1
ttengineconfig.selection=1

```

Figure 149. Secondary CLIFI Response File CID.RSP to Install Java 1.1.4

The component names are:

Runtime	Java Runtime
Unicode	Internationalization support that is part of the Java Runtime environment
Toolkit	Toolkit
TlktDoc	Toolkit Documentation

Samples	Samples
Debugger	OS/2 ICAT Debugger for Java
Unifont	Times New Roman MT 30 Unicode font
TTengine	Updated TrueType engine required for the Unicode font

Set `Selection=1` for each component you want to install, and set `Selection=0` for each component you do not want to install.

Note

There is a new `config` component associated with each installable component. The `config` selection variable must always be set to the same value as the selection variable for the component. For example, if you set `toolkit.selection=0`, you must also set `toolkitconfig.selection=0`.

The Runtime component is always installed in the `\JAVA11` directory on the target drive and is a prerequisite for the Toolkit, Samples, and Debugger components.

The Internationalization support portion of the Runtime component and the Times New Roman MT30 Unicode font component are always installed on the boot drive. If the Times New Roman MT30 Unicode font component is installed, the TrueType component is required.

The target drive and directory for the other components can be specified by setting the drive and path variable to the desired values. For example, to install the Debugger component in the `F:\JAVADEBUG` directory, modify the `CID.RSO` file as follows:

```
debugger.selection=1
debuggerconfig.selection=1
debugger.dbgdrv=F:
debugger.dbgpath=\JAVADEBUG
```

If a previous version of Java exists in the specified directory on the target drive, the installation program replaces it. If a previous version of a Java component was installed and you have not selected to reinstall that component, the installation program displays a window to warn you that this component will be downlevel and let you chose to have the component upgraded.

To suppress this confirmation window, along with others encountered during installation, set the `java11.continue` variable to `UNATTENDED`:

```
java11.continue=UNATTENDED
```

C.1.2 Starting the Unattended Installation

Execute the following single-line command for doing a CID installation of Java 1.1.4:

Java 1.14 CID Install Syntax

```
CLIFI /A:C  
      /R2:CID.RSP  
      /R:JAVA114.RSP  
      /B:C:  
      /S:X:\IMG\JAVA11  
      /L1:CIDERR.LOG  
      /L2:CIDHIST.LOG
```

where:

- /R Specifies the fully qualified location of the Feature Install response file (use JAVA114.RSP as delivered on the enclosed CD-ROM)
- /B Specifies the boot drive
- /S Specifies the fully qualified location of the extracted Java 1.1.4 for OS/2 Warp files
- /L1 Specifies the fully qualified location of the optional error log file
- /L2 Specifies the fully qualified location of the optional installation history log file

Tip: In addition to the two log files, also check the locally stored \OS2\INSTALL\WPINSTAL.LOG file on the boot drive if problems occur during installation.

This execution command is stored as CID.CMD and resides on the enclosed CD-ROM (\CID\EXE directory).

C.1.3 Performing an Unattended Uninstall

To uninstall Java 1.1.4 for OS/2 Warp, enter the following single-line command at an OS/2 command prompt:

```
CLIFI /A:U /F:"<WP_INSTALLED>" /O:INV_JAVA11 /SET:Selection=ALL
```

C.2 Installing Netscape 2.02 Refresh Unattendedly

If you plan to have Netscape 2.02 work with Java 1.1.4, we recommend installing Java 1.1.4 prior to Netscape. If this is not done, you can later run a command, NSJAVA.EXE, to switch to Java 1.1.4. This command lets you also toggle between Java 1.02 and Java 1.1.4.

C.2.1 Creating the Netscape Response File

The following keywords are available to allow you to install Netscape Navigator for OS/2 in a CID environment.

COMP	= Netscape Navigator
FILE	= C:\NETSCAPE
AUX1	= C:
CFGUPDATE	= AUTO
DELETEBACKUP	= NO
OVERWRITE	= YES
SAVEBACKUP	= NO
NSCONVERTBROWSER	= YES
NSCONVERTQL	= YES
NSDELALL	= NO
NSDELMAIL	= NO
NSDELNEWS	= NO
NSDELSEC	= NO
NSDELINI	= NO
NSDELBOOKMARKS	= NO
NSJAVA11	= YES

Figure 150. Response File for Unattended Installation of Netscape

where:

COMP Required keyword. Specifies the component to be installed (used by Software Installer). In this example, Netscape is to be installed.

The value given here must exactly match the `NAME` keyword of the `COMPONENT` entry (see about line 90 in the `NETSCAPE.PKG` file).

FILE	<p>Required keyword. Specifies the fully qualified path where Netscape is to be installed.</p> <p>If, in the response file, you specify the drive or directory of an existing Netscape installation (including an installation for another platform, such as Microsoft Windows), the existing installation will be overwritten.</p> <p>Example:</p> <pre>FILE = D:\NETSCAPE</pre>
AUX1	<p>Required keyword. Specifies the drive where the JAVAOS2 directory and tree is created.</p> <p>Files on an existing JAVAOS2 directory for the specified drive will be overwritten.</p> <p>Example:</p> <pre>AUX1 = C:</pre>
CFGUPDATE	<p>Required keyword. Specifies whether or not to update the configuration files, such as CONFIG.SYS and AUTOEXEC.BAT. We recommend setting this keyword to Auto. Possible values are Yes, No, and Auto.</p> <p>Example:</p> <pre>CFGUPDATE = AUTO</pre>
DELETEBACKUP	<p>Required keyword. Specifies whether or not to delete the product when a deinstallation takes place. The possible values are Yes or No.</p> <p>Example:</p> <pre>DELETEBACKUP = NO</pre>
OVERWRITE	<p>Required keyword. Specifies whether or not to overwrite existing files. The possible values are Yes or No.</p> <p>Example:</p> <pre>OVERWRITE = YES</pre>

SAVEBACKUP	Required keyword. Specifies whether or not a backup version of the current product is to be created when the product is updated. The possible values are <i>Yes</i> or <i>No</i> . Example: SAVEBACKUP = NO
NSCONVERTBROWSER	Required keyword. Specifies whether or not to change the default browser for all URL objects to be the Netscape Navigator for OS/2 (This function applies to OS/2 Warp 4 only!). Example: NSCONVERTBROWSER = YES
NSCONVERTQL	Required keyword. Specifies whether or not to convert the Web Explorer QuickList to Netscape Navigator bookmarks. Example: NSCONVERTQL = YES
NSDELALL	Optional keyword. Specifies whether or not to delete all of the following keywords: NSDELMAIL, NSDELNEWS, NSDELSEC, NSDELINI, and NSDELBOOKMARKS.
NSDELMAIL	Optional keyword. Specifies whether or not to delete all mail messages.
NSDELNEWS	Optional keyword. Specifies whether or not to delete all news items.
NSDELSEC	Optional keyword. Specifies whether or not to delete all security certificates.
NSDELINI	Optional keyword. Specifies whether or not to delete all settings (NETSCAPE.INI).
NSDELBOOKMARKS	Optional keyword. Specifies whether or not to delete all bookmarks (BOOKMARK.HTM).
NSJAVA11	This keyword is optional and is introduced with the refresh version of Netscape 2.02. It specifies whether or not to switch to Java 1.1 support if Java 1.1 is installed.

Yes means switch to Java 1.1 support if installed.
No means switch to Java 1.02 support.

C.2.2 Initiating Remote Install

Software Installer is used to install Netscape. The following command line is an example of a response file installation of Netscape Navigator for OS/2:

```
INSTALL /X /A:I /NMSG /O:DRIVE /R:<response file> /L2:<output file>
```

For more information on the INSTALL.EXE command (Software Installer), enter the following command at an OS/2 command prompt in the directory where the unzipped files of Netscape 2.02 Refresh reside:

```
VIEW EPFIHELP.INF
```

C.3 Installing Netscape 2.02 OS/2 Plugins Unattendedly

The following response file allows the OS/2 Plug-in Pack to be installed in a CID environment.

```
COMP                = OS/2 Multimedia MPEG Support
COMP                = OS/2 Multimedia Plug-Ins for Netscape Navigator
COMP                = 16-bit Plug-In Support for Netscape Navigator
FILE                = C:\NETSCAPE
AUX1                = C:\MMOS2
CFGUPDATE           = AUTO
DELETEBACKUP        = NO
OVERWRITE           = YES
SAVEBACKUP          = NO
NAV_PRESENT         = TRUE
```

Figure 151. Response File for Unattended Installation of Netscape Plug-Ins

where:

COMP Required keyword. Specifies the component to be installed (used by Software Installer). In this example, Netscape is to be installed.

The values given here must exactly match the `NAME` keyword of the `COMPONENT` entry of the appropriate `.PKG` file (see line 2 in the `OS2MPEGC.PKG` file and see lines 31 and 271 in the `OS2PIP.PKG` file).

FILE	<p>Specifies the fully qualified path where Netscape is installed.</p> <p>If you don't know where Netscape is installed at the target workstation, you can omit this keyword. Software Installer checks the OS2.INI file if and where it is installed.</p> <p>Example:</p> <pre>FILE = D:\NETSCAPE</pre>
AUX1	<p>Specifies the drive where the OS/2 Multimedia (MMOS2) is installed.</p> <p>If you don't know where MMOS2 is installed at the target workstation, you can omit this keyword. Software installer checks the PATH statement where MMOS2 is installed.</p> <p>Example:</p> <pre>AUX1 = C:\MMOS2</pre>
CFGUPDATE	<p>Required keyword. Specifies whether or not to update the configuration files, such as CONFIG.SYS and AUTOEXEC.BAT. We recommend setting this keyword to Auto. Possible values are Yes, No, and Auto.</p> <p>Example:</p> <pre>CFGUPDATE = AUTO</pre>
DELETEBACKUP	<p>Required keyword. Specifies whether or not to delete the product when a deinstallation takes place. The possible values are Yes or No.</p> <p>Example:</p> <pre>DELETEBACKUP = NO</pre>
OVERWRITE	<p>Required keyword. Specifies whether or not to overwrite existing files. The possible values are Yes or No.</p> <p>Example:</p> <pre>OVERWRITE = YES</pre>

SAVEBACKUP Required keyword. Specifies whether or not a backup version of the current product is to be created when the product is updated. The possible values are Yes or No.

Example:

```
SAVEBACKUP = NO
```

NAV_PRESENT Required keyword. Specifies whether or not Netscape Navigator for OS/2 has already been installed.

TRUE means Netscape is installed.

FALSE means Netscape is not installed.

Example:

```
NAV_PRESENT = TRUE
```

C.3.1 Initiating Remote Install

Software Installer is used to install the OS/2 Plug-ins of Netscape. The following command line is an example of a response file installation of the Netscape Navigator Plug-ins for OS/2:

```
INSTALL /X /A:I /NMSG /O:DRIVE /R:<response file> /L2:<output file>
```

For more information on the INSTALL.EXE command (Software Installer), enter the following command at an OS/2 command prompt in the directory where the unzipped files of Netscape 2.02 Plug-ins reside:

```
VIEW EPFIHELP.INF
```


Appendix D. Special Notices

This redbook describes the CID (Configuration, Installation and Distribution) enablement of OS/2 Warp 4 with all its subcomponents. It is also a handbook that provides step-by-step guidance in all phases of the usage and administration of the OS/2 base operating system, such as SEINST and CLIFI (Command Line Interface Feature Installer, introduced with OS/2 Warp 4 for the first time), its subcomponents and main OS/2 products for the implementation of CID in an OS/2 LAN environment using LCU (LAN CID Utility), NetView DM/2 or TME 10 SD 3.1.3.

This document is intended for workstation specialists and system technical personnel responsible for mass distribution of OS/2 products in an OS/2 Warp 4 LAN. Some knowledge of LAN redirection principles and TCP/IP is assumed. This redbook solely focuses on OS/2 Warp 4. If CID-related information is needed for previous OS/2 versions, such as OS/2 Warp Connect or even OS/2 V2.11, or software distribution techniques other than mentioned above, you need to obtain the redbook titled *OS/2 Installation Techniques: The CID Guide*, SG24-4295.

This redbook gives a broad understanding of a new features introduced with OS/2 Warp 4 that were not available with previous OS/2 versions. It is essential to know about these new features, such as Feature Installer (FI), to successfully make use of them and distribute OS/2 Warp 4.

References in this publication 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 program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX ®	Application System ®
AS/400 ®	CICS ®
Client Access	Client Access/400 ®
CT ®	DB2 ®
Extended Services ®	FFST
First Failure Support Technology	IBM ®
LAN Distance ®	Micro Channel ®
MQ	MVS ® (logo)
NetView ®	Operating System/2 ®
OS/2 ®	Presentation Manager ®
PS/2 ®	ThinkPad ®
Workplace Shell ®	XGA ®

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications, see "How To Get ITSO Redbooks" on page 519.

- *OS/2 Installation Techniques: The CID Guide*, SG24-4295
- *Network Clients for OS/2 Warp Server: OS/2 Warp 4, DOS/Windows, Windows 95/NT, and Apple Macintosh*, SG24-2009
- *OS/2 Warp Server, Windows NT, and NetWare: A Network Operating System Study*, SG24-4786
- *Inside OS/2 Warp Server, Volume 1: Exploring the Core Components*, SG24-4602
- *Inside OS/2 Warp Server, Volume 2: System Management, Backup/Restore, Advanced Print Services*, SG24-4702
- *OS/2 Warp Server Functional Enhancements: Part 1*, SG24-2008
- *Getting to Know OS/2 Warp 4*, SG24-4758
- *Prepare for OS/2 Engineer Certification*, SG24-4869

E.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

E.3 Other Publications

These publications are also relevant as further information sources:

- *NetView Distribution Manager /2 Version 2.1: Change Distribution Manager User's Guide*, SH19-5048
- *Getting to Know OS/2 Warp 4*, ISBN 0-13-842147-1
- *Prepare for OS/2 Engineer Certification*, ISBN 9-655-61119-1

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL <http://www.redbooks.ibm.com>.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** – to order hardcopies in United States
- **GOPHER link to the Internet** – type GOPHER WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get lists of redbooks:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Home Page on the World Wide Web**

<http://w3.itso.ibm.com/redbooks>

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pbl/pbl>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibm.link.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) – send orders to:

	IBMMAIL	Internet
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1) 408 256 5422 (Outside USA)** – ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services** – send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications	http://www.elink.ibm.link.ibm.com/pbl/pbl
Catalog	

- **Internet Listserver**

With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibm.link.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank).

List of Abbreviations

ABIOS	Advanced BIOS	DDP	Device Driver Profile
AS/400	Application System/400	DHCP	Dynamic Host Communications Protocol
APAR	Authorized Program Analysis Report	DLC	Data Link Control (SNA)
API	Application Programming Interface	DLL	Dynamic Link Library
ART	Application Registration Tool	DM	Distribution Manager
ASD	Alternative Software Delivery	DOS	Disk Operating System
BIOS	Basic Input Output System	DS	Distributed Services
CC	Change Control	EA	Extended Attributes
CD-ROM	(optical read) Compact Disk - Read Only Memory	EE	Extended Services
CDM	Change Distribution Manager	FFST	First Failure Support Technology
CICS	Customer Information Control System	FI	Feature Installer
CID	Configuration, Installation, Distribution	GRADD	GRaphics Adapter Device Driver
CLIFI	Command Line Interface Feature Installer	HTTP	HyperText Transmission Protocol
CSD	Corrective Service Diskettes	IBM	International Business Machines Corporation
CSF	Corrective Service Facility	IEEE	Institute of Electrical and Electronics Engineers
DASD	Direct Access Storage Device	IEEE 802.2	LAN Data-Link Protocol (DLC Protocol)
DAX	Developer API eXtensions	IPX	Internetwork Packet eXchange (datagram)
DB2	Database 2	IDE	Integrated Drive Electronics
DBM	Database Manager	ITSO	International Technical Support Organization
DDI	Device Driver Installation	LAA	Locally Administered Address
		LAN	Local Area Network
		LAPS	LAN Adapter and Protocol Support

LBA	Logic Block Addressing	PCI	Peripheral Component Interconnect
LCU	LAN CID Utility	PM	Presentation Manager
LDU	LAN Download Utility	PPP	Point-to-Point Protocol
LLC	Logical Link Control (LAN, top sublayer of layer 2, IEEE 802.2)	PSP	Personal Software Products divisionI
MCA	Micro Channel Architecture	RAM	Random Access Memory
MPTS	Muti-Protocol and Transport Services	RDT	Rapid Deployment Team
MVS	Multiple Virtual Storage	REXX	Restructured EXtended eXecutor Language
NBDD	NetBIOS Datagram Distributor	RIPL	Remote Initial Program Load
NBNS	NetBIOS Name Server	RMPI	Remote Printer Installation
NC	Network Computing	ROM	Read Only Memory
NCSD	Network Computing Software Division	RSU	Remote Software Update
NDIS	Network Driver Interface Specification	RUI	Request Unit Interface (SNA)
NETBEUI	NetBIOS Extended User Interface	SD	Software Distribution
NetBIOS	Network Basic Input Output System	SDM	Software Distribution Manager
NFS	Network File System	SLIP	Serial Line Interface Protocol
NIC	Network Interface Card	SNA	Systems Network Architecture
NIF	Network Information File	SNA/DS	Systems Network Architecture/Distributed Services
NTS	Network Tele Systems	SWIM	Software Installation and Maintenance
NTS/2	Network Transport Services for OS/2	SrvIFS	Server Installable File System
NVDM	NetView Distribution Manager	TCP	Transmission Control Protocol
NVDM/2	NetView Distribution Manager for OS/2		
Open32	Developer API Extensions; also known as DAX		
OS/2	Operating System/2		

TCP/IP	Transmission Control Protocol/Internet Protocol
TCPBEUI	TCP NetBIOS Extended User Interface
TME	Tivoli Management Environment
URL	Universal Resource Locator
VDM	Virtual DOS Machine
VGA	Video Graphics Array
WAN	Wide Area Network
WIN-OS/2	Windows environment in OS/2
WPS	Workplace Shell
WWW	World Wide Web

Index

A

abbreviations 523
ABIOS 484
acronyms 523
Alternative Software Delivery 154
ALT-F1 40
ANXCLT.INI 409
ANXIFS 301, 406
ANXSRV.INI 409
APPC 176, 342
ARCHIVE 64
ASD 154, 156, 370
ASD Requirements 157
ASD-PreReq 156
Attended 13

B

BACKUP 65
BASE.CMD 334
bibliography 517
BIOS 40, 485
boot diskettes 287, 319
BUNDLE 77

C

CASAGENT 75, 104
CASAGENT.EXE 53, 57
CASCKREX 75
CASDELETE 75
CASINSTL 75, 305
CASSETUP 21
Catalog 353
CC Server 340
CC server 358
CDBOOT.EXE 42
CDM 24, 339, 348
CDM ADD_WS 385
CDM catalog 361, 383
CDM INSTALL 394
CDM LIST_REQ 394
CDM LIST_WS 385, 394
CDM QUIESCE TRANSM 382
CDM SHUTDOWN 382
CDM START 382
CDM STATUS 351, 382

CDM STOP 382
CDROMINST 42
Change Control Server 340
Change Distribution Manager 339
Change Files 353
change profile 370
CICS 168, 171, 178
CID 3, 75
CID Structure 17
CID-enabled 3, 6, 311, 339, 383, 395
CLIFI 44, 75, 162, 204, 217, 254, 372, 412, 503
CLNDESK 75, 463
CMRECORD 275
CMSETUP 171, 274
COMMIT 72
Communications Manager/2 165
Communications Server for OS/2 Warp 165, 171
CONINST.EXE 42
CONTROL.INF 482
COPYFROMFLOPPY 42, 43
COPYFROMPFLOPPY 41
Corequisite Group 393
Corrective Service Facility 62
CSD 169, 345
CSF 62, 71
CSFSTAGEDRIVE 41

D

Database 2 for OS/2 166, 171, 177, 247, 343
DBADM 344
DDI 481, 484
DDIDDP 190
DEFAULT.CMD 53
DESKTOP 48
Desktop Icon Shuffler 121, 380
Desktop Shuffler 463
DHCP 114
DISKETTESOURCE 42
DISKTYPE 48
Display Driver Install 373
DRVMAP.INF 482
DSPINSTL 75, 89, 373
DSPINSTL.EXE 52
DUALBOOT 49

E

EXIT_WHEN_DONE 72
ExitOnError 182
EXITWHENDONE 42

F

FDISK 361, 485
FDISKETTESOURCE 48
Feature Install (CLIFI) Response File 254
Feature Installer 39, 155, 159, 216, 370, 503
FI 39, 75
FIBASE.RSP 44, 59, 216
FIRSTDISK 48
FISSETUP 370
FixPak 61, 62, 317, 345, 372
FORMAT 50
FSERVICE 41, 66
FSERVICE.EXE 63
FULLINST 49

G

GETBOOT 75, 96
GETOSCID 75, 78
GETREXX 75, 77, 96

H

HPFS 78
HPFSREQ 49

I

IBMIDECD.FLT 490
IBMNVD2.INI 307
IBMTOK.NIF 115
IDE 485
IFSDEL 75, 106
INSCFG32.DLL 77
INSTALL.DLL 52
INSTALL.EXE 52
INSTALL.INI 44
Installation by Replication 5
Installation keywords 9
Installation Modes
 Attended 13
Installed Features 155
INSTALLTYPE 48
INSTFFST.CMD 55
INT13 485

IPX 274

ISA 49

ISHIELD.EXE 44, 53, 54, 57

ISHUFFL 381

J

Java 1.1.4 503
JAVAOS2 508

K

KICKER 372

L

LAN Adapter and Protocol Support 22
LAN CID Utility 21, 22, 96, 106
LAN Distance Response File 240
LAN Download Utility 339, 340
LAPS 22
LAPSRSP 219, 222, 225
LCU 21, 57, 96, 305, 311, 313, 322
LDR.RSP 60
LDU 339, 340, 348
LDU Manager 348
Lightly attended 13
LOCAL.CMD 53, 54
Lotus Notes 169, 179
Lotus Notes OS/2 Client Response File 282
Lotus SmartSuite 170, 180
Lotus SmartSuite 96 Response File 283

M

MACHINE 49
Matrox Millenium 92
MBR 486
MFS.RSP 60
Microchannel 49
MMOS2 511
Mobile File Sync Response File 247
MODEL 49
MOUSECOM 49
MPTS 163, 364
MPTS Response File 219
MPTS Utilities 112, 164
MPTS.RSP 59
MPTSTART.CMD 295, 299
MQ Series for OS/2 171, 177
MQ Series Response File 250

MQSeries 168
MVS 340

N

NetBIOS 294, 341, 395, 403
NETFIN.RSP 60
NetFinity Response File 241
Netscape 2.02 OS/2 Plug-ins 503
Netscape 2.02 Refresh 503
NetView DM/2 11, 21, 24, 118, 169, 217, 339
NetWare 11
Netware Requester 274
Network SignOn Coordinator 376
Network Signon Coordinator/2 Response File 245
NFS 11, 303, 403, 405, 422, 492
Novell 11
NPCONFIG.EXE 42, 51, 57, 59
NPNWINST 56
NPNWINST.EXE 274
NPNWINST.RSP 60
NSC.RSP 60
NUMDISKS 48
NVDM BLD 432
NVDM ERASEREQ 421, 437
NVDM LSRQ 420
NVDM.CFG 402, 419
NVDM/2 306
NVDM/MVS 342
NVDM2.INI 345
NVDMBDSK 75, 118, 306, 354
NVDMINST 346
NVDMPPMS 345, 347
NWMPTS.RSP 60

O

OEMPROGRAM 42
OS/2 LAN Requester Response File 233
OS/2 Peer Response File 230
OS/2 Peer Services 377
OS/2 response file 182
OS/2 Warp Server 11, 167
OS2_SHELL 42
OS2IMAGE 163
OS2PEER.RSP 59

P

PCACODE 359

PCI 40, 342
PCMCIA 319, 414
PCOMOS2 176
PEERRMT 55
Personal Communication LITE 165
Personal Communications for OS/2 Warp 165, 171, 173
Personal Communications for OS2 Warp 378
Personal Communications Response File 277
PLANARID 49
PM_SPOOLER_QUEUE 483
PMSHELL 44
PRDESC.LST 209
Printer and Job Properties 147
PROTOCOL.INI 116, 219, 341
PROTSHELL 42
Pull Mode 36
Push Mode 36

R

RAMABIOS 48
Redirected Installation 10
Remote Software Update 61, 154
REPLACE_NEWER 72
REPLACE_PROTECTED 72
REQUIRED 77
RESERVE.SYS 492
RESOLV2 295, 430
Response files 8
response files 181
RESPONSE.FIL 63
RESPONSE.RSP 169, 282
Return Codes 80, 82, 84, 120, 322
REXX 311, 314, 322, 358
RINSPRN 131
RINSTCFG 75
RINSTPRN 75, 131, 143, 379, 482
RIPL 11
RSPDDI 481
RSPINST 41, 75, 84, 404
RSU 61, 154
RUNINSTALL 44, 50

S

SAMPLE.RSP 182
SAVECONNECT 42, 43
SBCD2.ADD 490
SCSI 50

SD4OS2 403, 407, 419
SEDISK 75, 76, 79, 287, 354, 404
SEIMAGE 75, 76, 79, 81, 404
SEINST 75, 76, 79, 82, 404
SEMAINT 76, 79, 84, 404
SERVICE.EXE 63
SERVICE.INI 104, 315, 320
ServicePak 62
SETBOOT.EXE 55
SETUP.CMD 296, 299
SHAREA 345
SHAREB 345
SHPIINST.DLL 77
SHUF_CLT.INI 125
SHUF_KEY.INI 125
SNOOP.LST 40, 290, 490
snooping 40
Software Choice 154, 503
Software Installer 150, 244
Software Installer Response File 244
SOURCEPATH 41, 48
SQLENSG 343
SRVIFS 11, 23, 98, 303, 315
SRVIFSC 102
SRVREXX 360
Standard Installation 3
STARTDBM 343
STARTNB.CMD 419
STARTUP.CDM 53
SUBMODEL 49
SUITE.RSP 170, 283
SVAGENT.RSP 60
SYSADM 344
SYSINST2 41
SYSLEVEL.OS2 62
SYSTEM_INI 48
SystemView Agent Response File 245

T

TARGETPATH 41, 48
TCP/IP 11, 114, 226, 236, 295, 303, 375, 395, 403, 421
TCP/IP Response File 236
TCPAPPS.RSP 59, 236
TCPBEUI 114, 226, 293, 298, 299
THINIFS 75, 101, 104, 303
THINLAPS 113, 115, 294, 295, 319, 354, 355
THINSRV 75, 99, 315

TME 10 SD 3.1.3 11, 21, 217, 395
Transaction Server (CICS) Response File 251
TrueType 505

U

UHPFS.DLL 78
Unattended 13
Unicode 504
Unifont 505
USER.RSP 50, 60
USER_INI 48
UserExit 202

V

VGA 52, 93
VIOADAPTER 49
VIODISPLAY 49
VIOMEM 49

W

WR08415 22
WRAPPER.EXE 53

ITSO Redbook Evaluation

The OS/2 Warp 4 CID Software Distribution Guide
SG24-2010-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)



Printed in U.S.A.