

OS/2 Installation Techniques: The CID Guide

Document Number SG24-4295-01

1996

International Technical Support Organization
Boca Raton Center

Take Note!

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

Second Edition (May 1996)

This edition applies to

- IBM Operating System/2 Warp Version 3
- IBM Operating System/2 Warp Connect
- IBM Multi-Protocol Transport Services Version 5.00
- IBM Operating System/2 Local Area Network Server V5.0
- IBM Communications Manager/2 Version 1.11
- IBM DATABASE 2 for OS/2 Version 2.11
- Novell NetWare Version 4.1
- IBM TCP/IP Version 3.0
- IBM NetView Distribution Manager/2 Version 2.1

and related products.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 91J Building 235-2 Internal Zip 4423
901 NW 51st Street
Boca Raton, Florida 33431-1328

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 1996. 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.

Abstract

The CID Guide describes the CID (Configuration Installation and Distribution) method and implementation. It is also a handbook that provides detailed step-by-step guidance in all phases of the usage and administration of the main products for the implementation of CID in an OS/2 LAN environment. It covers remote installations of OS/2 and CID-enabled products using *IBM OS/2 LAN Server V5.0 RIPL* or *LAN CID Utility* or *Novell NetWare* or *IBM TCP/IP Version 3.0* or *IBM NetView Distribution Manager/2*.

This document is intended for workstation specialists and system technical personnel responsible for mass distribution of CID-enabled software in an *OS/2 V2.x* or *OS/2 Warp V3* LAN. Some knowledge of LAN redirection principles is assumed. The CID redbooks GG24-3977, GG24-3780-02, GG24-3781-01, GG24-3783-01 and GG24-4295-00 are replaced by this document.

The included CDROM contains some CID-related programs and sample CID control files for the different installation scenarios. To support older program levels, it also contains machine readable versions of the replaced redbooks and their sample diskettes.

(657 pages)

Contents

Abstract	iii
Figures	xvii
Tables	xxi
Special Notices	xxiii
Preface	xxvii
How This Document is Organized	xxvii
Related Publications	xxxix
International Technical Support Organization Publications	xxxiv
How To Get ITSO Redbooks	xxxv
How IBM Employees Can Get ITSO Redbooks	xxxv
How Customers Can Get ITSO Redbooks	xxxvii
IBM Redbook Order Form	xxxix
Acknowledgments	xl

Part 1. General CID Overview and Introduction	1
Chapter 1. CID History, Concepts and Scenarios	3
1.1 Steps Towards CID	4
1.1.1 Standard Installation Method	4
1.1.2 Installation by Cloning	6
1.1.3 CID-Enabled Installation	8
1.2 CID Concepts (and Terminology)	10
1.2.1 Response Files	11
1.2.2 Redirected Installation (Redirected I/O or Drives)	13
1.2.3 Software Distribution Manager	19
1.3 Installation Modes	20
1.3.1 Attended Installation	20
1.3.2 Lightly Attended Installation	21
1.3.3 Unattended Installation	22
1.3.4 Clarification	22
1.3.5 Summary	23
1.4 Installation Scenarios	24
1.4.1 Installation on Workstations without Operating System	24
1.4.2 Installation/Upgrade on DOS/Windows Workstations	25

1.4.3	Installation/Upgrade on OS/2 Workstations	25
1.4.4	Installation of applications	26
1.5	Overview of Workstation-Based Software Distribution Managers	26
1.5.1	LAN CID Utility	26
1.5.2	NetView Distribution Manager/2	30
1.5.3	Positioning LAN CID Utility and NetView DM/2 Summary	31
1.6	Setup of a Code Server	32
1.6.1	CID Structure	33
1.6.2	Server Types	33
1.6.3	Manual Setup	33
1.6.4	Automated Setup	34
1.7	Installation Process	34
1.8	CID Enablement of Products	34
1.8.1	Response Files	35
1.8.2	Command Line Driven Execution	35
1.8.3	Redirected Drives	35
1.8.4	Progress Indication and Logging Facilities	35
1.8.5	Standard Return Codes	36
1.8.6	Transfer of Product Diskettes	36

Part 2. CID System Usage and Administration 37

Chapter 2. Recommended CID Directory Structure	39
2.1 CID Directory Structure Considerations	40
2.2 NetView DM/2 Directory Structure Considerations	43
Chapter 3. Response Files	47
3.1 Introduction to Response Files	47
3.1.1 Why Have a Response File?	47
3.2 Build Response Files	48
3.2.1 OS/2 Response File	48
3.2.2 Communications Manager/2	52
3.2.3 DATABASE 2 for OS/2 Response File	56
3.2.4 MPTS Response File	58
3.2.5 Creating Response Files for LAN Server	65
3.2.6 NetView DM/2 V2.1 Client Feature	74
3.2.7 TCP/IP Response File	75
3.3 Special Considerations	76
Chapter 4. Client Installation Control Files	79
4.1 CID Installation Commands	79
4.1.1 OS/2	79

4.1.2 LAN Adapter and Protocol Support	89
4.1.3 LAN CID Utility	94
4.1.4 Communications Manager/2	108
4.1.5 Personal Communications/3270 for OS/2	112
4.1.6 Communications Server for OS/2 Warp Version 4.0	116
4.1.7 Database 2 for OS/2	117
4.1.8 LAN Requester/Server	121
4.1.9 TCP/IP V3.0	123
4.1.10 Product Installation Order	127
4.2 Installation Commands for Products that Are Not CID-Enabled	128
4.2.1 Installation of Novell NetWare Workstation for OS/2 V2.01	128
4.3 Handling of Locked Files	138
4.3.1 Locked File Solution Using IBMLANLK.EXE	139
4.3.2 Locked File Handling Using NetView DM/2 V2.1	142
4.4 LCU Command File	143
4.4.1 LCU Overview	144
4.4.2 LCU Return Codes Processing	144
4.4.3 Working with Default Response Files and LCU Command Files	146
4.4.4 LCU Command File Structure	148
4.4.5 Adding Products to the LCU Command File	154
4.4.6 LCU Command File Execution of a Diskette Initiated Installation	156
4.4.7 The LCU Command File - Samples and Skeletons	163
4.5 Using LCU CASPREP Utility	169
4.6 NetView DM/2 Change Control Files	171
4.6.1 Objects, Global Names and NetView DM/2 Catalog	171
4.6.2 Change Files and Change File Profiles	173
4.6.3 Create Change Files to Install CID-Enabled Products	176
Chapter 5. Maintenance and Service	183
5.1 Connecting a Client for Maintenance	183
5.1.1 Using Boot Diskettes	183
5.1.2 Maintenance Partition	184
5.2 Introduction to Corrective Service Facility	188
5.2.2 Logging Information	193
5.2.3 Interrupted Service	194
5.3 Servicing of OS/2 Products	194
5.3.1 Service Pak and Select Pak	195
5.3.2 Private Fixes	196
5.3.3 OS/2 Warp V3 Fix Pak	196
5.3.4 MPTS Upgrade	199
5.3.5 IBM Communications Manager/2 Version 1.1 Upgrade	201
5.3.6 LAN Server V4.0 Service Pak	203
5.3.7 IBM NetView Distribution Manager/2 Version 2.1 CSD	205

Chapter 6. Recovery Recommendations	207
6.1 LCU Recovery Capability	211
6.1.1 Sample LCU REXX Command File with Recovery	212
6.1.2 REXX Recovery Sequence Preceding a Single CID-Enabled Product Installation	213
6.1.3 LCU Recovery for Major Failures	215
Chapter 7. Remote Multiple Printer Support	217
7.1 Introduction	217
7.2 Definitions for Remote Printer Installation	218
7.2.1 Definition of Local Printers and Queues	218
7.2.2 Definition of Printer and Job Properties	219
7.2.3 Definition of Network Queues	219
7.2.4 Definition of WIN-OS/2 Printers	220
7.2.5 Miscellaneous Definitions	220
7.3 Remote Printer Installation Program	220
7.4 Backup and Restore of Printer and Job Properties	224
7.4.1 Backup of Printer and Job Properties	224
7.4.2 Restoring Printer and Job Properties	225
7.4.3 Holding or Releasing a Queue	225
7.5 Printer Response File Keywords	226
7.5.1 Keywords Description	227
7.6 Sample Printer Response File	234
7.6.1 Response File Configurator	239
7.6.2 Printer Installation Sample	240
7.7 Using LCU	241
Chapter 8. Auto-Partitioning the Hard Disk	243
8.1 Multiple Fixed Disks	243
8.1.1 Restrictions	243
8.2 The Fixed Disk Utility Program (FDISK)	244
8.2.1 Description of FDISK for OS/2	244
8.2.2 Functions on Options Pull-Down Menu of FDISK	246
8.3 FDISK Command Line Interface	247
8.3.1 The Command Parameter of the FDISK Command Line	247
8.3.2 The Restrictor Parameter of the FDISK Command Line	248
8.3.3 Output Created by FDISK	250
8.4 The OS/2 SETBOOT Command	251
8.5 The Sample REXX Partitioning Utilities	251
8.5.1 The FDISK 'FILE:' Parameter	252
8.5.2 PIPE.CMD	252
8.6 Using REXX Code to Create Partitions on Hard Disks	253
8.6.1 Accessing REXX from a Client Workstation	253

8.6.2 Summary of DISKPREP.CMD Function	254
8.7 Disk Partitioning when Using NVDM/2	255
8.7.1 Write a Batch Procedure without REXX for NVDM/2	257

Part 3. CID System Generation and Administration	261
---	------------

Chapter 9. Hardware and Software Requirements	263
9.1 Hardware	263
9.1.1 Server: Base Hardware	263
9.1.2 Server: Hard Disk Drives	263
9.1.3 Clients	265
9.2 Software	266
9.2.1 Servers	266
9.2.2 Common Requirements	266
9.2.3 Clients	267

Chapter 10. Manual Setup of IBM Operating System/2 Local Area Network Server V5.0 RIPL	269
10.1 Overview of Remote Initial Program Load (RIPL)	270
10.2 Overview of Installation Steps for RIPL	271
10.3 Manual Installation of LAN Server V5.0 RIPL Code Server	273
10.3.1 Preparation of Basic Code on the Code Server	273
10.3.2 Creating the CID Directory Structure	274
10.3.3 Loading OS/2 CID Utilities to Code Server	275
10.3.4 Copy Files from the Sample Code CDROM to Code Server	275
10.3.5 Loading Diskette Images	275
10.3.6 Copy REXX to Code Server	276
10.3.7 Copy SETBOOT.EXE and XCOPY.EXE to Code Server	276
10.3.8 Code Server Installation.	276
10.3.9 Build Response Files	286
10.3.10 Build Client Installation Control Files	286
10.4 Preparation for RIPL Clients	286
10.4.1 Preparation of the RIPL Installation Diskette	287
10.4.2 Test CID RIPL Installation to One Client	288
10.4.3 Create "Remote IPL Workstation" Definitions for Each CID Client	289
10.5 Running the Code Server	290
10.5.1 Starting the Code Server	290
10.5.2 Stopping the Code Server	290
10.5.3 Display Code Server Status	290
10.6 Customizing the Code Server	291
10.6.1 Code Server Security	291

10.6.2	Working with Authorizations and Client Workstation Names . . .	291
Chapter 11.	Manual Setup of LAN CID Utility	293
11.1	IBM Multi-Protocol Transport Services Overview	293
11.2	LAN CID Utility Overview	294
11.3	Scenario	296
11.4	Manual Installation	297
11.4.1	Basic Installation of Code Server	297
11.4.2	Creating the CID Directory Structure	297
11.4.3	Loading OS/2 CID Utilities to Code Server	297
11.4.4	Copy Files from the Sample Code CDROM to Code Server . . .	298
11.4.5	Loading Diskette Images	298
11.4.6	Copy REXX to Code Server	298
11.4.7	Copy SETBOOT.EXE and XCOPY.EXE to Code Server	298
11.4.8	Code Server Installation (THINSRV)	299
11.4.9	Build Response Files	302
11.4.10	Build Client Installation Control Files	302
11.5	Preparation of Client Workstations	302
11.5.1	Running SEDISK	302
11.5.2	Adding LAN Transport System to Client Diskettes (THINLAPS) .	302
11.5.3	Install LCU Redirector (THINIFS)	305
11.5.4	Install LCU client (CASINSTL)	307
11.6	Running the Code Server (SERVICE.EXE)	310
11.6.1	Starting the Code Server	311
11.6.2	Stopping the Code Server	311
11.6.3	Display Code Server Status	311
11.6.4	Refresh Authorization List of Code Server	312
11.6.5	Forcing Code Server to Stop	312
11.7	Customizing the Code Server	312
11.7.1	Code Server Security	312
11.7.2	Customizing Client Diskettes	314
11.7.3	Customizing the SERVICE.INI file	315
Chapter 12.	Manual Setup of Novell NetWare	317
12.1	Overview	317
12.2	Scenario	318
12.3	Manual Installation	319
12.3.1	Basic Installation of NetWare Server and NetWare Requester .	319
12.3.2	Creating the CID Directory Structure	319
12.3.3	Loading OS/2 CID Utilities to Code Server	320
12.3.4	Loading Diskette Images	320
12.3.5	Copy REXX to Code Server	320
12.3.6	Copy SETBOOT.EXE to Code Server	320

12.3.7	Code Server Installation	321
12.3.8	Build Response Files	321
12.3.9	Build Client Installation Control Files	322
12.4	Preparation of Client Workstations	322
12.4.1	Running SEDISK	322
12.4.2	Adding LAN Transport System to Client diskettes	322
12.5	Running the Code Server	327
12.5.1	Customizing Client Diskettes	327
Chapter 13.	Manual Setup of IBM TCP/IP Version 3.0	329
13.1	Overview	329
13.2	Scenario	330
13.3	Manual Installation	330
13.3.1	Basic Installation of TCP/IP Server	331
13.3.2	Creating the CID Directory Structure	331
13.3.3	Loading OS/2 CID Utilities to Code Server	332
13.3.4	Loading Diskette Images	332
13.3.5	Copy REXX to Code Server	333
13.3.6	Copy SETBOOT.EXE to Code Server	333
13.3.7	Code Server Installation	333
13.3.8	Build Response Files	334
13.3.9	Build Client Installation Control Files	335
13.4	Preparation of Client Workstations	335
13.4.1	Running SEDISK	335
13.4.2	Adding LAN Transport System to Client Diskettes	335
13.5	Running the Code Server	339
13.5.1	Customizing Client Diskettes	339
13.6	Additional Installation Procedures and Files	340
Chapter 14.	Manual Setup of NetView Distribution Manager/2	349
14.1	NetView DM/2 Overview	349
14.2	Scenario	351
14.3	Overview of Installation Steps for NetView Distribution Manager/2 CC server	351
14.4	Manual Installation	352
14.4.1	Preparation of Basic Code on the Code Server	352
14.4.2	Creating the CID Directory Structure	352
14.4.3	Loading Diskette Images	352
14.4.4	Loading OS/2 CID Utilities to Code Server	353
14.4.5	Code Server Installation	353
14.4.6	Build Response Files	356
14.4.7	Build Change Files	356
14.5	Preparation of Client Workstations	356

14.5.1	Creation of Boot Diskettes	356
14.5.2	Connectivity between CC Server and CC Client	361
14.6	CC Client Operational States	363
14.7	Install Change Files on Client Workstations	364
14.8	Operating the Code Server	367
14.8.1	NetView DM/2 Command Interface	367
14.9	Problem Determination	368
14.10	Customizing the Code Server	370
14.10.1	User Profile Management (UPM) Considerations	370
14.10.2	NetBIOS Considerations	371
Chapter 15.	OS/2 CID Utilities	373
15.1.1	Loading OS/2 CID Utilities to the Code Server	373
15.1.2	SEDISK	377
Chapter 16.	Loading Product Images to Code Server	379
16.1.1	Loading OS/2 Diskette Images with SEIMAGE	379
16.1.2	Loading LAN Transport System Diskette Image(s) with LAPSDISK	382
16.1.3	Loading Communications Manager/2 Diskette Images	384
16.1.4	Loading DB2/2 Diskette Images	385
16.1.5	Loading LAN Server Diskette Images	386
16.1.6	Loading NetView DM/2 Diskette Images	391
16.1.7	Loading TCP/IP Diskette Images	392
16.1.8	Loading LAN CID Utility Files	394
16.1.9	Loading SRVIFS Files	394
16.1.10	Loading NetWare Requester Files	395
Chapter 17.	LAN CID Utility	397
17.1.1	GETBOOT	397
17.1.2	GETREXX	398
17.2	Quick Reference	400
Chapter 18.	Automated Setup with CASSETUP	403
18.1	Functions of CASSETUP	403
18.2	Requirements	404
Chapter 19.	Migration and How to Add New Products	405
19.1	Code Server Migration	405
19.2	Migrating a Code Server from NTS/2 LCU to MPTS LCU	406
19.3	Migrating a Code Server from LCU to NetView DM/2	406
19.4	Adding New Products	407

Part 4. CID Enabling of Applications	409
Chapter 20. Software Installer	411
20.1 Files created by Software Installer	411
20.2 Install Parameters	412
20.3 Default Response File	413
20.4 Example for a Product that uses Software Installer	414
20.5 Additional Information	415
20.6 Enabling a New product to Software Installer	416
20.6.1 Step by Step Description	417
20.7 Example for non CID-Enabled Software	423
Part 5. Appendixes	425
Appendix A. File Index Table	427
Appendix B. Versions Used in This Book	431
Appendix C. OS/2 Response File Keywords	433
C.1 Keyword Description	436
C.2 How to Edit the Response File	461
C.3 Printer Description Table for AdditionalPrinters and DefaultPrinter	
Keywords	462
C.4 Response Files Basics	463
C.4.1 Response File Processing	464
C.4.2 Group and Client Response Files	464
C.5 Response File Syntax	466
C.5.1 Comment Lines	466
C.5.2 Response Lines	467
C.5.3 Keywords	467
C.5.4 Values	468
C.5.5 Including Other Response Files	470
C.6 Response File Style Recommendations	470
C.6.1 Keyword Guidelines	470
C.6.2 Standard Keywords	471
C.6.3 Order of Response File Execution	473
C.6.4 List Implementation	474
C.7 Response File Processing Sequence	474
C.7.1 Response File Hierarchy	474
C.8 Include Keywords, Detailed Explanation	476
C.8.1 Single Include and IncludeAtEnd File Example	476

C.8.2 Single IncludeInLine File Example	477
C.8.3 Multiple Include Files Example	478
C.9 The User Exit Keywords of the Response File	480
Appendix D. OS/2 V2.1 CID Installation Utility for SVGA Adapters	483
D.1 Usage of SVGA CID Support	483
D.2 Detailed Description of the Utilities	485
Appendix E. LAN Network Adapters	489
E.1 NDIS 2.0.1 MAC Drivers	489
E.2 Network Interface Card Support Matrix for OS/2 Warp V3 LAN Systems	512
E.3 LS Product Matrix	523
E.3.1 Support for Additional Drivers	534
Appendix F. Create Environment Variables Program Description	545
F.1 How to Use CRENVVAR.EXE	545
F.2 Samples with CRENVVAR.EXE	546
F.3 Make File, DEF File and Source Code for CRENVVAR.EXE	547
F.3.1 Make File for CRENVVAR C Routine	547
F.3.2 CRENVVAR.DEF Define File	547
F.3.3 C Source for CRENVVAR.C	547
Appendix G. Use of Other Code Servers	551
G.1 LAN File Services/ESA	552
G.1.1 CID Installation	554
G.2 LAN Resource Extension and Services/VM	554
G.2.1 CID Installation	557
G.3 IBM Client Access for AS/400	558
G.3.1 Connectivity Alternatives	560
G.3.2 CID Installation	560
Appendix H. Sample Code Diskette/CDROM	563
Appendix I. Hardware and Software Dependencies	571
I.1 OS/2 Versions and CID	571
I.1.1 OS/2 Warp V3 and OS/2 Warp with WinOS2 V3	571
I.2 Loadable BIOS and CID	573
I.3 PCMCIA and CID	576
I.4 RAID and CID	581
Appendix J. CID Enabled Applications	583
J.1 IBM Products	583

J.2 Independent Software Vendor Products	589
Appendix K. CID Installation Messages and Return Codes	595
K.1 OS/2 CID Utilities' Error Messages	595
K.2 RSPINST Return Codes	596
K.3 IFSDEL CID Return Codes	601
K.4 Architected CID Return Codes	602
Appendix L. The SERVICE.INI File Keywords	605
L.1 NETBIOS resources	611
L.2 The Use of Redirected Drives with Aliases	612
Appendix M. DISKPREP.CMD	617
Glossary	627
List of Abbreviations	633
Index	635

Figures

1. Standard Installation Method	5
2. Cloning an Installation	7
3. CID-Enabled Installation	9
4. Using A Response File	12
5. Response File Example	13
6. Redirected Installation	14
7. Redirected Drive	15
8. MPTS LCU SDM Constellation	27
9. The CID Directory Structure	41
10. The NetView DM/2 Directory Structure	45
11. LAN Server V5.0 Installation and Configuration Panel	66
12. LAN Server V5.0 Configure Window	70
13. Extract of an LCU Client Command File Illustrating SEINST Program Invocation	81
14. Extract of an LCU Client Command File Illustrating SEMAINT Program Invocation	89
15. Extract of an LCU Client Command File Illustrating MPTS Program Invocation	92
16. Extract of an LCU Client Command File Illustrating THINIFS Program Invocation	99
17. Extract of an LCU Client Command File Illustrating IFSDEL Program Invocation	101
18. Extract of an LCU Client Command File Illustrating CASINSTL Program Invocation	105
19. Extract of an LCU Client Command File Illustrating CASDELETE Program Invocation	108
20. Extract of an LCU Client Command File Illustrating CMSETUP Program Invocation	111
21. Extract of an LCU Client Command File Illustrating INSTALL Program Invocation	115
22. Extract of an LCU Client Command File for CM Server Install	117
23. Extract of an LCU Client Command File Illustrating DBCID Program Invocation	121
24. Extract of an LCU Client Command File Illustrating LANINSTR Program Invocation	123
25. Extract of an LCU Client Command File Illustrating INSTALL Program Invocation	127
26. NWINST.CMD Procedure	130
27. NWDELETE.CMD Procedure	133
28. NWSEED.CMD Procedure	134

29.	STARTNW.CMD File	134
30.	NWPREP.CMD Procedure	136
31.	Modifications in CONFIG.SYS at First Reboot	161
32.	ASCII Change File Profile for OS/2 Warp Connect	177
33.	Sample Service Pak Response File	193
34.	The Extended LCU Directory Structure for Service	194
35.	OS/2 Warp V3 Fix Pak Response File	197
36.	LCU Command File Using SRVIFS for OS/2 Warp V3 and Fix Pak Install	198
37.	Product Definition Part for OS/2 Warp V3 Service Pak	199
38.	Response File for MPTS CSD WRx8150	200
39.	Product Definition Part for MPTS CSD WRx8150	200
40.	MPTS CSD WRx8150 Install Section	201
41.	CM/2 V1.11 Upgrade Response File	202
42.	Product Definition Part for CM/2 V1.11	202
43.	CM/2 V1.11 Install Section for an Upgrade	203
44.	LAN Server V4.0 Service Pak Response File	204
45.	Product Definition Part for LAN Server V4.0 Service Pak IPx8152	204
46.	LAN Server V4.0 Install Section for CSD IPx8152	205
47.	LCU REXX Command File with Recovery	212
48.	Sample Recovery REXX Statements	214
49.	Sample Workstation Printer Scenario (OS/2 Part)	236
50.	Sample Workstation Printer Scenario (WIN-OS/2 Part)	237
51.	Sample Printer Response File	238
52.	PRINTERS.CMD to Be Called from UserExit	240
53.	Sample Output from FDISK Command Line Query	250
54.	Example FDSK.RSP File	252
55.	PIPE.CMD Program	252
56.	SRVAUTO.PRO File	278
57.	Overview of LAN Server V5.0 GUI.	279
58.	RIPL template menu.	280
59.	First page of RIPL notebook.	281
60.	Parameters for RIPL Client.	282
61.	General information about RIPL Client.	283
62.	Changes of the CIDMODEL.FIT File	284
63.	CONFIG.30 of the Master Workstation	284
64.	STARTRPL.CMD File	285
65.	STARTUP.CMD File	286
66.	RIPL Installation AUTOEXEC.BAT	288
67.	LAN CID Utility Environment Using SRVIFS	296
68.	PROTOCOL.INI File of the LTS Diskette after MPTS THINLAPS	303
69.	CONFIG.SYS File of the LTS Diskette after MPTS THINLAPS	304

70.	Last Part of CONFIG.SYS File of the LTS Diskette after Second THINIFS	307
71.	CONFIG.SYS File of the LTS Diskette after CASINSTL	308
72.	STARTUP.CMD File of the LTS Diskette Created by CASINSTL	309
73.	Relation between AuthList and IFS Statement	313
74.	Modified CONFIG.SYS of Second LTS Diskette	325
75.	STARTRFI.CMD File	326
76.	ENV_VARS.CMD File for NetWare	327
77.	Example EXPORTS File on NFS TCP/IP Server for CID	333
78.	Example HOSTS File	334
79.	Modified CONFIG.SYS of Second LTS Diskette	337
80.	NFSRFI.CMD File	338
81.	ENV_VARS.CMD File for TCP/IP	339
82.	THINTCP.CMD Procedure	341
83.	TCPCOPY.CMD Procedure	344
84.	TCDELETE.CMD Procedure	345
85.	TCPSEED.CMD Procedure	346
86.	TCPREP.CMD Procedure	347
87.	Stand-Alone LAN Environment	350
88.	CONFIG.SYS on "NVDM/2 DISK #1" after NVDMBDSK	360
89.	NetView DM/2 Client Directory Structure on NetView DM/2 Boot Disk #1	361
90.	Changefile profile for command line	369
91.	OS/2 Warp V3 SEIMAGE Directory Structure	380
92.	LAN Server V4.0 Installation Task Window	387
93.	LAN Server V4.0 Copy Product Diskettes Window	388
94.	LAN Server V5.0 CID Subdirectory Structure	390
95.	NetView DM/2 V2.1 Profile for IBM Works	414
96.	LCU product definition sequence for IBM Works	415
97.	IBMWORKS Response file.	415
98.	Software Installer folder.	416
99.	Software Installer sample product folder.	416
100.	Software Installer install utility folder.	417
101.	Software Installer Enabling new applications: Step1.	418
102.	Software Installer Enabling new applications: Step2.	419
103.	Software Installer Enabling new applications: Step3.	419
104.	Software Installer Enabling new applications: Step4.	420
105.	Software Installer Enabling new applications: Step5.	421
106.	Software Installer Enabling new applications: Step6.	421
107.	Software Installer Enabling new applications: Step7.	422
108.	Printer Description List	463
109.	Response File Multiple Include Pattern	480
110.	Product Definition Part for the SVGA Utilities	484

111.	Installation Part for the SVGA Utilities	485
112.	NTS/2 LAPS PROTOCOL.INI	538
113.	Sample NTS/2 LAPS Response File TRCCA.RSP for IBM Token-Ring 16/4 Credit Card Adapter	541
114.	CID Installation Using Host DASD	551
115.	CID Installation Using Host DASD	554
116.	Services Provided by LANRES	557
117.	CID Installation Using Host DASD	558
118.	CID Installation Using Host DASD	561
119.	The Sample Code CDROM Directory Structure	563
120.	Loadable ABIOS Installation Scenarios	574
121.	Sample SERVICE.INI	611
122.	Drive Redirections Using Alias under SRVIFS	614

Tables

1.	Basic Characteristics of the Standard Installation Method	6
2.	Basic Characteristics of the Cloning Method	8
3.	Basic Characteristics of CID-Enabled Installations	10
4.	Installation Modes (Summary)	23
5.	Positioning LAN CID Utility and NetView DM/2	31
6.	Overview of SVGA Adapters and Their Corresponding .DSC Files	84
7.	TCP/IP V3.0 INSTALL Parameters	124
8.	Product Variable Descriptions	150
9.	Hardware Specifications	243
10.	Disk Space Recommendations for Diskette Images	264
11.	Basic Software for Code Server	266
12.	Security by Use of AuthList Keyword and Client Workstation Name	314
13.	CC Client State	363
14.	TCP/IP V2.0 Abbreviated Names	392
15.	Remote Installation of OS/2 Command Quick Reference	400
16.	File Index Table	427
17.	OS/2 Response File Keywords Table	433
18.	Overview of Manufacturing Codes for SVGA Adapters	486
19.	Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1	489
20.	Network Interface Card Support for OS/2 Warp LAN Systems	513
21.	Network Interface Card Support for LAN Systems	524
22.	OS/2 Versions and CID	571
23.	OS/2 V2.0 and Machine Types	575
24.	Results of PCMCIAOptions settings in OS/2 response file	579
25.	Results of PCMCIAOptions settings in OS/2 response file	579
26.	Results of PCMCIAOptions settings in OS/2 response file	580
27.	CID Enabled IBM Products	583
28.	CID Enabled Products by Other Vendors	589

Special Notices

This publication is intended to help customer technical personnel and IBM system engineers install software and manage changes in a LAN network. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM OS/2, IBM LAN Adapter and Protocol Support, IBM Network Transport Services/2, IBM Multi-Protocol Transport Services, IBM LAN Server, IBM Communications Manager/2, IBM DATABASE 2 for OS/2, IBM TCP/IP and IBM NetView Distribution Manager/2. See the PUBLICATIONS section of the IBM Programming Announcement for these products for more information about what publications are considered to be product documentation.

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, 208 Harbor Drive, Stamford, CT 06904 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

DATABASE 2	DB2/2
DISTRIBUTED DATABASE CONNECTION SERVICES/2	FFST/2
First Failure Support Technology/2	IBM
LANStreamer	Micro Channel
NetView	Operating System/2
OS/2	Personal System/2
Presentation Manager	PS/ValuePoint
PS/2	RISC System/6000
ThinkPad	Ultimedia
ValuePoint	WIN-OS/2
Workplace Shell	XGA

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

ATI	ATI Technologies, Inc.
Cirrus Logic	Cirrus Logic, Inc.
DCA	Digital Communications Associates, Inc.
EtherCard PLUS	Standard Microsystems Corporation
EtherLink	3COM Corporation
Headland Technology	Headland Technology Inc.
IRMAtrac	Digital Communications Associates, Inc.
Logitech.	Lotus Development Corporation.
Microsoft, Windows, Word	Microsoft Corporation
NIU	Ungermann-Bass, Inc.
NetWare	Novell, Inc.
Novell	Novell, Inc.
PostScript	Adobe Systems, Inc.
Standard Microsystems	Standard Microsystems Corporation
S3	S3 Incorporated
Trident	Trident Microsystems, Incorporated
TSENG	Tseng Laboratories, Inc.
Ungermann-Bass	Ungermann-Bass, Inc.
Weitek	Weitek Corporation
Western Digital	Western Digital Corporation
1-2-3, Lotus, Freelance, Freelance Graphics	3Com

3Com Corporation	386
Intel Corporation	486
Intel Corporation	

Preface

This publication is intended to be the base reference guide for CID (Configuration, Installation and Distribution) management of software in the LAN environment. It is divided into five parts for different categories of readers.

The first part is an overview of CID that should enable the reader to understand the concept, methods and implementation. It contains no references to the rest of the book and should be regarded as an introduction to everyone, whether the person intends to use CID or not.

The second part is intended to be a reference for technical people that will use and administer a running CID system. It introduces the different types of response and control files that are the main means of control of the installation and update process on the clients.

The third part contains the information needed to create a CID code server for any of the LAN transport systems that support CID. Any information needed in this part already covered in part two is covered by references and not repeated. This part is only for the technician responsible for setting up the CID system.

The fourth part contains information on enabling applications for CID, including a chapter on Software Installer.

The fifth part contains appendixes with information and tables which are referenced from parts two, and three.

How This Document is Organized

The document is organized as follows:

- **Part 1: General CID Overview and Introduction**

This first part should be read by anyone who wants to understand the CID concept. It is the prerequisite for all other parts of this document but it does not contain any forward references to these other parts.

- **Chapter 1: CID History, Concepts and Scenarios** This chapter will give you conceptual knowledge about CID to create a base for understanding the following parts of this book. Products implementing these concepts are also briefly introduced.

- **Part 2: CID System Usage and Administration**

Part two is intended for the administrator of a running CID system. This is the person responsible for helping the clients by preparing the response and control files which are referenced when the client machine software is installed.

- **Chapter 2: Recommended CID Directory Structure** This section defines the recommended CID directory structure for the CID code server. Differences between LAN CID Utility (LCU) and NetView DM/2 are considered.
- **Chapter 3: Response Files** This chapter explains the reason for response files and also shows how to construct response files for all the main products.
- **Chapter 4: Client Installation Control Files** This section handles the CID installation utility commands and the control files these commands are using. The intricate workings of the *LCU Command File* and the *NetView DM/2 Change Control Files* are thoroughly explained.
- **Chapter 5: Maintenance and Service** The question about how to update a running CID system is covered in this chapter. How to update the code server and how to apply corrective service, service paks and new releases to the clients are also covered.
- **Chapter 6: Recovery Recommendations** This chapter tells what to do if something goes wrong during CID install. It shows the recovery capabilities of LCU and gives good advice.
- **Chapter 7: Remote Multiple Printer Support** A remote multiple printer installation program (RINSTPRN) is supplied with the book. This chapter explains how to use it.
- **Chapter 8: Auto-Partitioning the Hard Disk** This part provides information about the Fixed Disk Utility and shows some sample applications used to automate the partitioning of a hard disk.

- **Part 3: CID System Generation and Administration**

Part three is intended for the administrator responsible for constructing the CID system. This is the person responsible for building the CID code server(s) with the LAN transport system and all source images.

- **Chapter 9: Hardware and Software Requirements** This chapter specifies the recommended minimum configurations for code servers and client machines.

- **Chapter 10: Manual Setup of IBM Operating System/2 Local Area Network Server Version 3.0 RIPL** This section shows how to establish redirected drives for installation on a client using the RIPL feature of IBM Operating System/2 Local Area Network Server Version 3.0.
- **Chapter 11: Manual Setup of LAN CID Utility** Manual setup using the LAN CID Utility provided by the IBM Network Transport Services/2 is covered.
- **Chapter 12: Manual Setup of Novell NetWare** This chapter describes the setup of a Novell NetWare code server to use for remote install using the CID installation methods.
- **Chapter 13: Manual Setup of IBM TCP/IP Version 2.0** This section shows the setup of a TCP/IP server to install OS/2 and other CID enabled products on remote clients.
- **Chapter 14: Manual Setup of NetView Distribution Manager/2** This section describes the series of steps required to enable automated installation of CID-enabled products using IBM NetView Distribution Manager/2 Version 2.0 or higher.
- **Chapter 15: OS/2 CID Utilities** This chapter shows how to load the OS/2 CID Utilities into the CID server.
- **Chapter 16: Loading Product Images to Code Server** This part explains how to load the product images for some of the main CID enabled products into the code server.
- **Chapter 17: LAN CID Utility** Most of the functions of LAN CID Utility provided by the IBM Network Transport Services/2 are described in the context where they are used earlier in the book. The rest are described here.
- **Chapter 18: Automated Setup with CASSETUP** This chapter describes the CASSETUP program which assists the administrator in preparing the code server. It has been distributed with IBM Network Transport Services/2, but the latest version comes with MPTS/2.
- **Chapter 19: Migration and How to Add New Products** This section discusses how to migrate a code server to a higher level of LAN transport support or how to migrate from IBM Network Transport Services/2 to IBM NetView Distribution Manager/2 Version 2.0. It also gives advice about how to add new products to the code server.
- **Part 4: CID Enabling of Applications**

This part contains information on enabling applications for CID, including a chapter on Software Installer.

- **Chapter 20: Automated Setup with SRVSETUP** Software Installer is an IBM product that supports software developers with a set of programs and functions for developing installation programs. This chapter describes the use of Software Installer to allow a standardized way to install software products, and support manual and automatic software distribution and installation.

- **Part 5: Appendices**

The appendixes contain all tables, listings and reference material that is common for the previous parts of the document

- **Appendix A: File Index Table** This table is designed to be a quick reference to where a specific file can be found and where in the book there is a description on how to use it.
- **Appendix B: Versions Used in this Book** The listing shows the various software versions we used to test all installations described in this book.
- **Appendix C: OS/2 Response File Keywords** This part contains a description of all the keywords that can be used in an OS/2 response file. The table at the beginning shows which version of OS/2 they are valid for.
- **Appendix D: OS/2 V2.1 CID Installation Utility for SVGA Adapters** This appendix describes the utilities that enable remote installation and configuration of SVGA adapters.
- **Appendix E: LAN Network Adapters** The table contains network adapter driver descriptions, device driver file names, associated NIF file names and message file names for network adapter drivers distributed with the different LAN support products.
- **Appendix F: Create Environment Variables (CRENVVAR) Program Description** The CRENVVAR program is described with the source code. It is used in the installation procedures for Novell NetWare requester and IBM TCP/IP Version 2.0.
- **Appendix G: Use of Other Code Servers** This appendix introduces how to use CID when different types of host machines are used as code servers.
- **Appendix H: Sample Code Diskette/CDROM** This is the file/directory structure of the CD-ROM delivered with the book. The CD-ROM contains machine readable versions of the earlier CID redbooks and images of the sample diskettes. It also contains an image of the new sample diskette.

- **Appendix I: Hardware and Software Dependencies** This part shows some hardware and software dependencies the administrator should be aware of in order to successfully install OS/2.
- **Appendix J: CID Enabled Applications** These lists give an overview of which IBM and Independent Software Vendor (ISV) products and applications are CID-enabled.
- **Appendix K: CID Installation Messages and Return Codes** All the messages and return codes we have found for the OS/2 CID utilities are presented here. Also the architected CID return codes expected by the Software Distribution Managers (SDMs) are discussed.
- **Appendix L: The SERVICE.INI File Keywords** This is a description of the parameters used in the SRVIFS code server .INI file.
- **Appendix M: DISKPREP.CMD** This is a listing of the DISKPREP.CMD.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

IBM Communications Manager/2 Publications

- *IBM Communications Manager/2 Version 1.0 Network Administration and Subsystem Management Guide*, SC31-6168-00, available in softcopy only
- *IBM Communications Manager/2 Version 1.1 Network Administration and Subsystem Management Guide*, SC31-6168-01
- *IBM Communications Manager/2 Version 1.0 Workstation Installation Guide*, SC31-6169
- *IBM Communications Manager/2 Version 1.1 Workstation, Installation and Configuration Guide*, SC31-7169

IBM DATABASE 2 for OS/2 Publications

- *DATABASE 2 OS/2 Installation Guide*, S62G-3664
- *DATABASE 2 OS/2 Information and Planning Guide*, S62G-3662
- *DATABASE 2 for OS/2 Messages and Problem Determination Guide*, S62G-3668

IBM LAN Server V3.0 Publications

- *IBM Operating System/2 Local Area Network Server Version 3.0 Network Administrator Reference Volume 1 Planning and Installation*, S96F-8428
- *IBM Operating System/2 Local Area Network Server Version 3.0 Network Administrator Reference Volume 2 Performance Tuning*, S96F-8429
- *IBM Operating System/2 Local Area Network Server Version 3.0 Network Administrator Reference Volume 3 Network Administrator Tasks*, S96F-8430
- *IBM Operating System/2 Local Area Network Server Version 3.0 Productivity Aids*, S59G-4684

IBM LAN Server V4.0 Publications

- *IBM Operating System/2 Local Area Network Server Version 4.0 Network Administrator Reference Volume 1 Planning, Installation and Configuration*, S10H-9680
- *IBM Operating System/2 Local Area Network Server Version 4.0 Network Administrator Reference Volume 2 Performance Tuning*, S10H-9681
- *IBM Operating System/2 Local Area Network Server Version 4.0 Network Administrator Reference Volume 3 Network Administrator Tasks*, S10H-9682
- *IBM Operating System/2 Local Area Network Server Version 4.0 Commands and Utilities*, S10H-9686
- *Experiences with OS/2 LAN Server V4.0*, GG24-4428 (will be published later)
- *Automating OS/2 LAN Server V4.0 Administration*, GG24-4442 (will be published later)

IBM Network Transport Services/2 Publications

- *IBM Network Transport Services/2 LAN Adapter and Protocol Support Configuration Guide*, S96F-8489
- *IBM Network Transport Services/2 Redirected Installation and Configuration Guide*, S96F-8488

IBM Multi-Protocol Transport Services Publications

- *MPTS Configuration Guide*, S10H-9693
- *LAN CID Utility Guide*, S10H-9742

IBM NetView Distribution Manager/2 Publications

- *IBM NetView Distribution Manager/2 Version 2.1 Installation and Customization Guide*, SH19-6915-05
- *IBM NetView Distribution Manager/2 Version 2.1 User's Guide*, SH19-5048-02
- *IBM NetView Distribution Manager/2 Version 2.1 Messages and Error Recovery Guide*, SH19-6924-05

IBM NetView Distribution Manager Release 6 Publications

- *NetView Distribution Manager General Information*, GH19-6792-04
- *NetView Distribution Manager Application Programming Release 4*, SH19-6796-02
- *NetView Distribution Manager User's Guide*, SH19-6795-04
- *NetView Distribution Manager Installation and Customization Release 4*, SH19-6794-04
- *NetView Distribution Manager Overview and Scenarios*, SH19-6797-04

IBM Operating System/2 Publications

- *OS/2 Warp Connect Command Reference, Online Information*
- *OS/2 Warp Connect Operating System Installation Guide*
- *OS/2 Warp Connect Operating System Information and Planning Guide*

IBM TCP/IP V2.0 Publications

- *IBM Transmission Control Protocol/Internet Protocol Version 2 for OS/2: Installation and Administration*, SC31-6075-06
- *IBM Transmission Control Protocol/Internet Protocol Version 2.0 for OS/2: Network File System Guide*, SC31-7069-01

IBM Software Installer Publications

- *Software Installer*, SC34-4515
- *Examples using Software Installer*, GG24-2529

Personal Communications/3270 for OS/2 Publications

- *Personal Communications/3270 for OS/2 Version 4.0*, G221-4361

Other Publications

- *CID Enablement of DOS Local Area Networks*, SC31-6833
- *CID Enablement Guidelines*, S10H-9666-01
- *OS/2 System Software Distribution & Installation Using NetView DM/2*, GG66-3253
- *Software Profile Management Facility MVS/ESA Implementation Guide*, SC30-3574

International Technical Support Organization Publications

- *Communications Manager/2 Version 1.1 Enhancements*, GG24-4142
- *ValuePoint Systems*, GG24-4298
- *ThinkPad Systems*, GG24-4297
- *OS/2 Warp Version 3 and BonusPak "Exploring a New Generation"*, GG24-4426
- *OS/2 Warp Generation, Volume 1: OS/2 Warp V3, OS/2 Warp Connect, and Bonus Pak*, SG24-4552
- *The Guide to OS/2 Warp Device Drivers*, SG24-4627

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

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/redbooks>.

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 WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
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/redbooks.html>

- **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.

- **ITSO4USA 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)

IBMMAIL — send orders to:

In United States:	usib6fpl at ibmmail
In Canada:	caibmbkz at ibmmail
Outside North America:	bookshop at dkibmbsh at ibmmail

Internet — send orders to:

In United States:	usib6fpl@ibmmail.com
In Canada:	lmannix@vnet.ibm.com
Outside North America:	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	44-4th Avenue, S.W.	Sortemosevej 21
4800 Falls of the Neuse Road	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27609	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada (toll free)	1-800-267-4455
Outside North America	(+45) 48 14 2207
(long distance charge)	

- **1-800-IBM-4FAX (United States) or (+1) 415 855 43 29 (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

- **Redbooks Home Page on the World Wide Web**

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

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pbl/pbl>

- **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).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- Please put me on the mailing list for updated versions of the IBM Redbook Catalog.

First name	Last name
------------	-----------

Company

Address

City	Postal code	Country
------	-------------	---------

Telephone number	Telefax number	VAT number
------------------	----------------	------------

- Invoice to customer number

- Credit card number

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa.
Payment by credit card not available in all countries.
Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Acknowledgments

The advisors for the **1996 Update Edition** were:

Tom Arbuckle
International Technical Support Organization, Boca Raton Center

The authors of this update are:

Saad F. Al-Khaldi
Aramco Saudi Arabia

Martin Bense
GAD Gesellschaft fuer automatische Datenverarbeitung eG
Muenster, Germany

Ursula Kieserg
IBM Germany

Christof Steinmetz
IBM Germany

This update is the result of a residency conducted at the International Technical Support Organization, Boca Raton Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Alex Gregor
International Technical Support Organization, Boca Raton Center

Randy Baker
IBM Austin - Laboratory

Tim Moore
IBM Austin - Laboratory

Hank Thoresz
IBM Austin - Laboratory

Michael T. Ward
IBM Austin - Laboratory

The advisors for the **November, 1994 Edition** were:

Alex Gregor
International Technical Support Organization, Boca Raton Center

Eddie Griborn
International Technical Support Organization, Boca Raton Center

The authors of this document are:

Luis Ennser
IBM Brazil

Ursula Kieserg
IBM Germany

Terri Knott
IBM Canada

Bernhard Tenholte
IBM Germany

Anna-Lena Wretman
IBM Sweden

This publication is the result of a residency conducted at the International Technical Support Organization, Boca Raton Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Claude Bechard
International Technical Support Organization, Raleigh Center

Wolfgang Geiger
International Technical Support Organization, Raleigh Center

Randy Baker
IBM Austin - Laboratory

Jim Bostick
IBM Boca Raton - Laboratory

Roc Conder
IBM Boca Raton - Laboratory

John Feller
IBM Raleigh - Laboratory

John Howard
IBM Boca Raton - Laboratory

Salil Kulkarni
IBM Boca Raton - Laboratory

Timothy W. Moore
IBM Boca Raton - Laboratory

Larry Sullenger
IBM Austin - Laboratory

Michael T. Ward
IBM Austin - Laboratory

Part 1. General CID Overview and Introduction

This part provides basic knowledge about the CID architecture and terminology used in this book. It is meant as a general introduction to the subject that proves reliable independent of specific product versions.

Chapter 1. CID History, Concepts and Scenarios

The number of workstations installed in an organization has grown steadily over the past ten years. During that time, operating systems and application software have become larger and more complex. The process of installing software and data files required by the end users has now become an obstacle preventing installation of new systems and the upgrading of existing systems. The process of installing and maintaining OS/2* and subsystem products by diskette can be tedious and time consuming. In addition, many installation programs require data such as configuration information to be supplied at installation time. End users with little or no systems knowledge must not be required or expected to be involved in this process.

On the other hand, it is not feasible for an enterprise to have skilled systems personnel present every time a workstation needs to have software installed, configured or maintained. The increased connectivity that is available with OS/2 today can now be used to the advantage of the software administrator in an organization.

OS/2 and future IBM products have been (and will be) designed with the above requirements in mind, as IBM has designed a method to automate these processes, using redirected input/output on LAN-based client/server systems. This method was announced in October 1992 and was named: **Configuration, Installation, and Distribution**. We will use the acronym **CID**, to reference the products' capabilities of automated installation.

The primary goals of CID are to:

- Eliminate human intervention at the target workstation when preparing and executing the configuration, installation, migration and maintenance processes that are necessary to operate this workstation.
- Enable the code executing at the target workstation to perform all required configuration and installation tasks including the integration of previous customizations.
- Provide the capability to centralize human intervention to an administrator at a central preparation site.

In order to evaluate this software distribution and installation process, we will look at the work needed for standard manual installations, and then briefly look at an automated install process called cloning. After that we will introduce the concept of CID-enabled installations, helping you to:

1. Understand what CID is and how it works

2. Understand the administrator's tasks
3. Understand the CID process in detail depending on the products used and installed
4. Decide which product to use for managing a CID environment
5. Describe how to configure the individual workstations
6. Install and configure the CID Code Server

1.1 Steps Towards CID

This part of the book will give you conceptual knowledge about CID to create a base for understanding the following parts of this book. Products implementing these concepts are also briefly introduced.

Throughout this part we will not reference to any detailed information, as this will be an introduction for your general information concerning CID.

1.1.1 Standard Installation Method

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

- Human intervention is required to customize the product by passing configuration information to the installation program via 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, 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 step, 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 by computer systems.

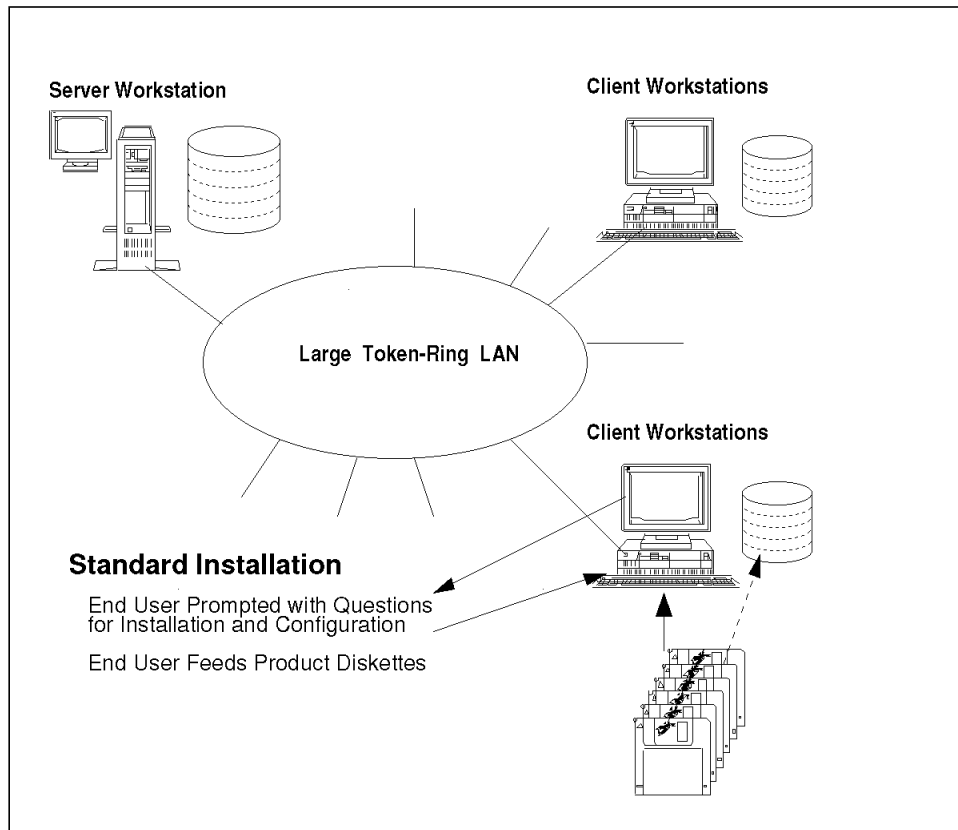


Figure 1. Standard Installation Method

As shown in Figure 1 an installation program is used that has been shipped with the product and that 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.

Standard installation scenarios, such as the above, always require that the end user provides installation and configuration information at the time of product installation. Thus, product configuration and product installation are not individual subtasks. Therefore, installation and configuration information must be provided at the same time by the same person at the place where the workstation is located. In other words, this installation method is not feasible if the installation process needs to be remotely managed. In addition, skilled people need to be present at the workstation location to perform this standard installation process.

The following table briefly summarizes the basic characteristics of the standard installation method:

<i>Table 1. Basic Characteristics of the Standard Installation Method</i>	
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

1.1.2 Installation by Cloning

To bypass the drawback of not being able to remotely manage a standard installation, a simple installation technique was developed. This installation method, known as *cloning*, executes an installation procedure previously written by an administrator. This procedure is built by performing the following steps:

- Install the product on the administrator's workstation in the same way you would customize and install it at the target workstation using the installation program delivered with the product.
- Discover the steps that were performed by the installation program (such as which files have been installed, which existing files have been updated with what kind of data) in order to create your own nondialog driven installation procedure. This is a very laborious, error-prone process since something has to be discovered that was intentionally hidden. This process is known as *reverse engineering* and is required to create a command line driven installation procedure that achieves the same result as the original installation program. The original installation program cannot be used to install the product at the target workstation if it requires dialog-driven configuration.

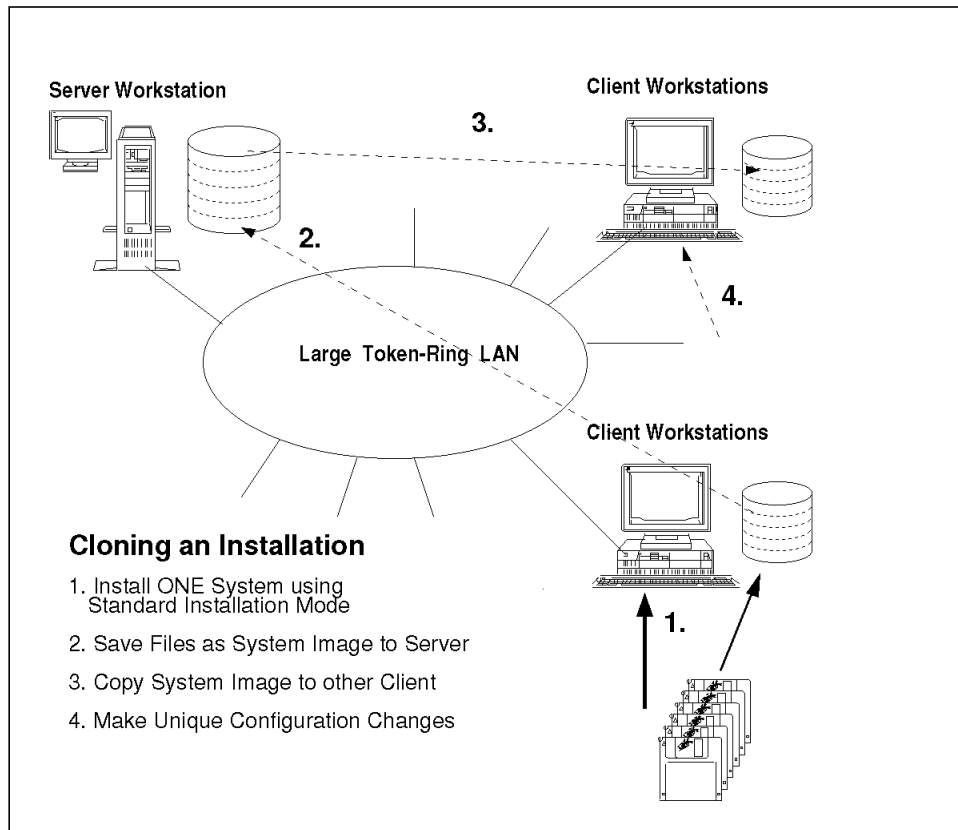


Figure 2. Cloning an Installation

While achieving the goal of minimizing human intervention at the target workstation by centralizing installation tasks at the administration site, cloning introduces drawbacks such as having to reverse engineer the installation process and to sort out the configuration dependencies between the administrator and the target workstation. In addition cloning does not migrate previous customizations.

The following table briefly summarizes the basic characteristics of the cloning method:

<i>Table 2. Basic Characteristics of the Cloning Method</i>	
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.	—

1.1.3 CID-Enabled Installation

In order to combine the strengths of both previously mentioned installation methods, *CID-enabled installation* has been developed with the basic objective of performing remote unattended installations of system software. It addresses the goals listed below by implementing the use of the product's original installation program at the target workstation as well as the capability to invoke and manage the installation process from a central site.

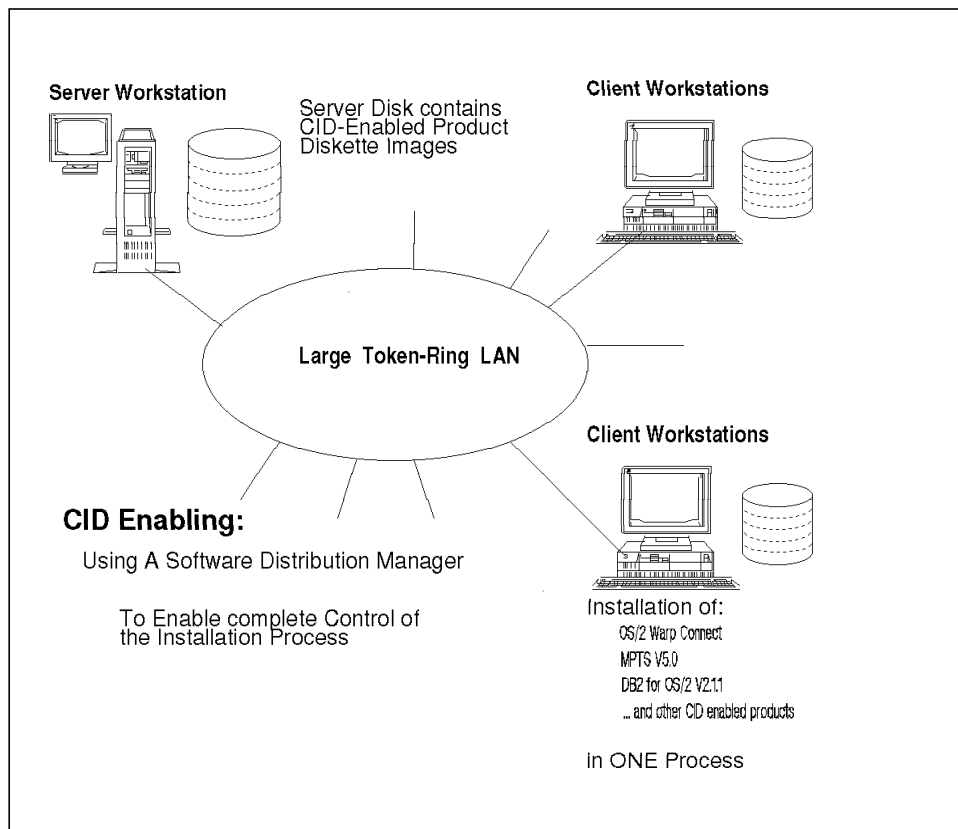


Figure 3. CID-Enabled Installation

- The product's specific installation program may be used for both locally managed as well as remotely managed installation processes. The logic of the installation program can be used to check underlying hardware and software to determine which code modules are to be installed and which files need to be updated.
- The time to specify installation and configuration information is no longer bound to the time when the installation process is executed. This allows the preparation and installation processes to be divided into two individually executable subtasks.
- Information which an installation program normally prompts for may be specified by a central administrator. 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 eliminate human intervention normally required to exchange media (such as diskettes), the product's code "images" must be transferred from the original medium to a medium that is large enough to store the entire image of code. This preparation step is performed before the installation process is started. During the installation process these *diskette images* are accessed.
- A facility must be provided to remotely control the installation process and to check whether the installation completed successfully. If required, the workstation will then automatically be rebooted.

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.

The following table briefly summarizes the basic characteristics of CID-enabled installations:

<i>Table 3. Basic Characteristics of CID-Enabled Installations</i>	
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.

1.2 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 program and data files to various clients. However, many of these solutions operate on the *cloning* principle, for example, the LAN Download Utility, which is a feature of IBM NetView Distribution Manager/2 Version 2.1.

In the past, the cloning 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.

There are several other functions and capabilities associated with the CID implementation to enhance usability and administration. These will be introduced within the rest of this part on the following pages.

1.2.1 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.

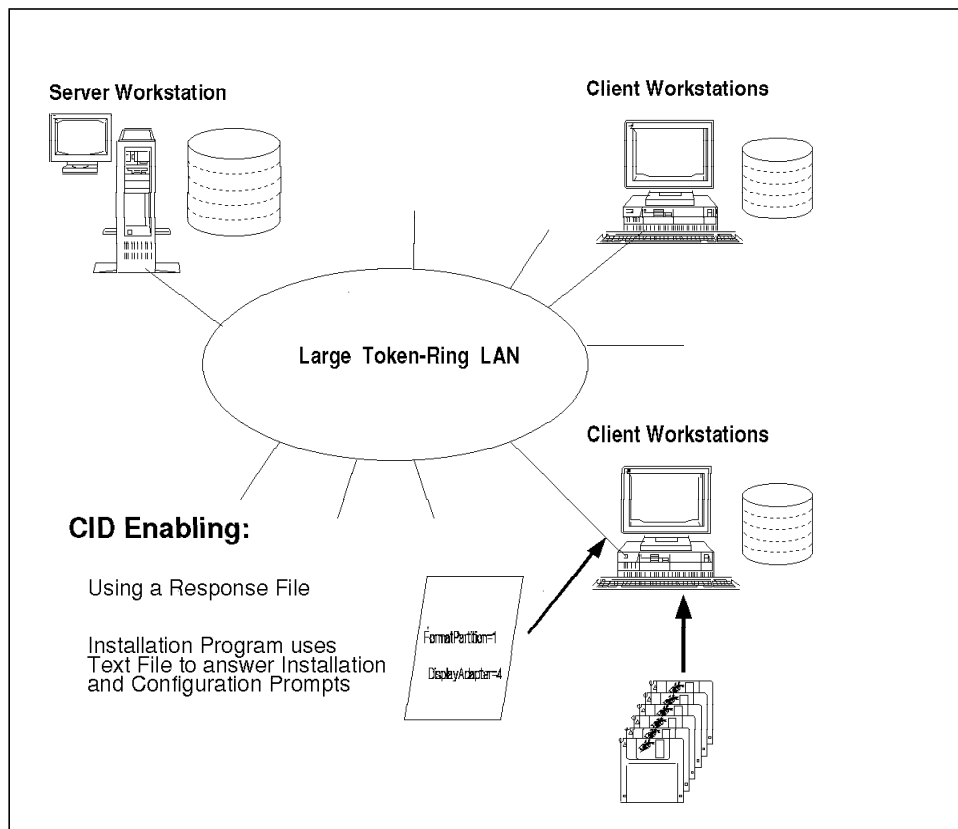


Figure 4. Using A Response File

Response files may include keywords which 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 5 on page 13 will show 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 5. Response File Example. Keywords and Values for disk formatting option and display selection of an OS/2 installation.

1.2.2 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:.

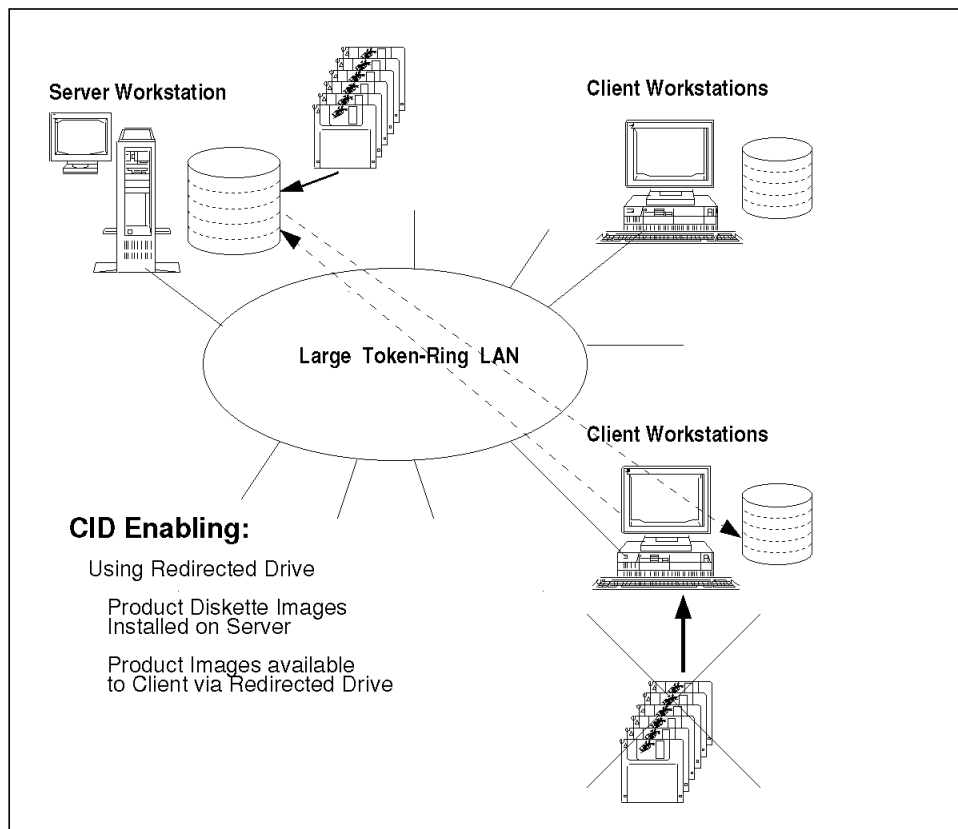


Figure 6. Redirected Installation

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 will be known as the **client** or **redirector** and the workstation that provides a remote (redirected) drive will be known as the **server** or **code server**.

The client workstations will access the drive on the server where the diskette 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**
- LAN Server V5.0 and Remote Initial Program Load (RIPL)
- NetView DM/2
- NetView DM for NetWare
- NetView DM/6000
- AS/400 based products

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

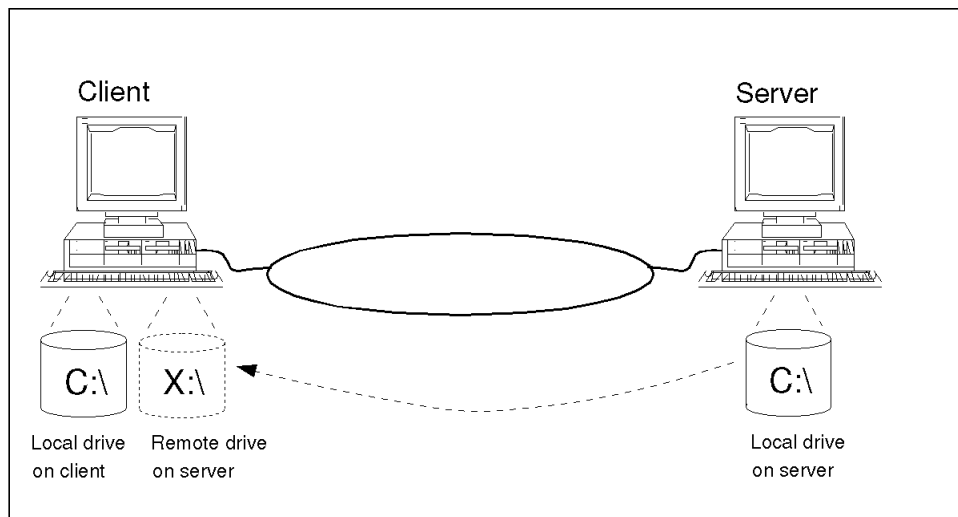


Figure 7. Redirected Drive

All descriptions in this book are based on IBM token-ring technology, NetBIOS, Novell IPX or TCP/IP protocols. If your network utilizes other LAN hardware, corresponding drivers are needed to establish hardware connections and low-level protocols. For detailed information about supported LAN hardware, see Appendix E, "LAN Network Adapters" on page 489.

Following is a brief explanation and positioning of the main connectivity bases.

1.2.2.1 SRVIFS Utility

SRVIFS is a small NetBIOS-based file server and requester, which is shipped with the IBM Multi-Protocol Transport Services (MPTS) product. The main use of SRVIFS is to provide redirected file I/O support to enable client access to a code server. This is a subset of the function provided by the LAN Server. Since SRVIFS requires a relatively small amount of disk space, it is particularly suited to being used during a boot diskette-based product installation.

1.2.2.2 TCP/IP NFS

TCP/IP provides a feature called Network File System (NFS) which can be used to share file resources across a network. Utilizing TCP/IP and NFS for redirected drive access provides a cross systems environment. This allows different system types running operating systems other than OS/2 to take on the role of code server. For example, a code server could be located on systems running TCP/IP on any of the following operating systems:

- OS/2
- AIX*
- VM
- MVS
- OS/400

If a code repository other than OS/2 was being used then the administrator would need to understand the consequences of that server not supporting extended attributes for OS/2 files.

1.2.2.3 Novell NetWare

The requester function of Novell NetWare (also called *NetWare Client*) could be used to install a complete operating system or individual CID-enabled products. The NetWare server must have the OS/2 support installed. To provide OS/2 extended attribute support, NetWare 3.11 or later is required.

Note

There is a space problem on the boot diskettes in this environment, because of an increase in size of the NetWare Client in recent versions. So, if you are about to set up a software distribution solution in a NetWare environment (both version 3.1x and 4.1), you should consider implementing a solution based on IBM NetView Distribution Manager for NetWare. See 1.2.2.6, "NetView DM for NetWare" on page 17

1.2.2.4 LAN Server V5.0 and Remote Initial Program Load (RIPL)

Remote Initial Program Load (RIPL) in conjunction with the requester function of LAN Server V5.0 can be used to install OS/2 Warp V3 on a system which has an unformatted fixed disk or a fixed disk that has not been partitioned or even on a system without a diskette drive installed. For a Remote Initial Program Load (RIPL) installation to take place the LAN adapter in the workstation must be enabled with a Remote Initial Program Load (RIPL) ROM module or the workstation must be started with a diskette containing the Remote Initial Program Load (RIPL) support. The connection to the redirected drives for the installation requires the file sharing services of the server function of LAN Server V5.0 to be active.

1.2.2.5 NetView DM/2

In a stand-alone LAN environment the connectivity is established using the IBM OS/2 NetBIOS support delivered with MPTS. When using NetView DM/2 as part of a host-connected environment you also have to configure the APPC function of Communications Manager/2 to establish additional connectivity to NetView Distribution Manager on the host, or to a remote administrator workstation, which is available since NetView DM/2 V2.1.

NetView DM/2 installs its own devices needed to create the redirection to the disk drives located on the server.

1.2.2.6 NetView DM for NetWare

IBM NetView Distribution Manager for NetWare Version 1.1 is another member of the NetView Distribution Manager family: The server part of the software resides on a NetWare V4.1 server (V3.1x is also supported). The client part of the software is called *NetView DM for NetWare Agent*

Agents are available for several operating systems:

- DOS
- DOS with MS Windows
- OS/2
- NetWare

In a LAN environment, connectivity between server and client is established using the IPX protocol of Novell NetWare. Like NetView DM/2 it is possible to connect a NetView DM for NetWare server to another NetView DM server (on an MVS host for example) using APPC communication. To provide the required APPC functionality you need NetWare for SAA V1.3 or V2.0 installed on this NetWare server.

Typically you may expect, that the tasks described in this Redbook using NetView DM for OS/2 can also be performed using NetView DM for NetWare or NetView DM/6000. For more information about NetView DM for NetWare, please refer to: *Software Distribution Using NetView Distribution Manager for NetWare*, GG24-4416.

1.2.2.7 NetView DM/6000

IBM NetView Distribution Manager/6000 Version 1.2 is another member of the NetView Distribution Manager family: The server part of the software resides on a RS/6000 server running AIX operating system. NetView DM/6000 and NetView DM for NetWare are basically the same code ported to a different target environment. So, there are only few differences, for example the available agents offer a larger selection including popular UNIX-like operating systems.

Agents are available for several operating systems:

- AIX
- DOS
- DOS with MS Windows
- OS/2
- several UNIX-like operating systems

In a LAN environment, connectivity between server and client is established using the TCP/IP protocol. Like NetView DM/2 it is possible to connect a NetView DM/6000 server to another NetView DM server (on an MVS host for example) using APPC communication. To provide the required APPC functionality you need IBM SNA Services/6000 installed on this AIX server.

Typically you may expect, that the tasks described in this Redbook using NetView DM for OS/2 can also be performed using NetView DM for NetWare or NetView DM/6000. For more information about NetView DM/6000, please refer to *NetView Distribution Manager/6000 Cookbook*, GG24-4246 and *NetView Distribution Manager/6000 Release 1.2 Agents and Advanced Scenarios*, GG24-4490.

1.2.2.8 AS/400 based products

There are also AS/400 based products that can be used for software distribution to workstations in a LAN. They provide similar capabilities like the other software distribution products mentioned above.

- Client Access for AS/400
- Managed System Services/400 (MSS/400)

- LAN Server functionality can also be provided by an AS/400

For further information see Appendix G, “Use of Other Code Servers” on page 551 and G.3, “ IBM Client Access for AS/400” on page 558.

1.2.3 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 will be 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* or *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:

- Not only is human intervention at the target workstation 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)
- But also 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 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, which 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.

1.3 Installation Modes

Although this document uses the term "installation", we should point out that this includes migration and maintenance as well. Unless otherwise noted, the CID enablement of the products discussed 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

1.3.1 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, 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 around to the end user for installation.
- You don't have to maintain all of these sets of diskettes.

Both 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.

1.3.2 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 two 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.

1.3.3 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.

1.3.4 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.

1.3.5 Summary

Table 4 summarizes the characteristics of the installation modes:

<i>Table 4. Installation Modes (Summary)</i>				
Installation Modes	Diskettes		Redirected Drives	
	Dialog-Driven Installation and Configuration	Response File	Dialog-Driven Installation and Configuration	Response File
Attended	Standard installation User initiates, feeds diskettes, provides responses	n/a	CID redirection only User initiates, provides responses	n/a
Lightly attended	n/a	CID response file only User initiates, feeds diskettes	n/a	CID User initiates
Unattended	n/a	n/a	n/a	Full CID No end user required NetView DM required
Note: n/a: not applicable				

1.4 Installation Scenarios

This section describes different general installation scenarios depending on the environment of the client workstation to be installed. We will limit the scope of this chapter to the description of OS/2 operating system installation scenarios.

Each installation scenario is dependent on the client workstation environment before the actual installation.

The client workstation environment can be:

- Workstations with NO operating system installed
- Upgrading DOS or DOS/Windows** systems to OS/2 Warp Connect
- Upgrading OS/2 V2.x systems to OS/2 Warp Connect or OS/2 Warp Connect with WinOS2 (according to the appropriate upgrade path)
 - systems without WinOS/2 support (also known as "OS/2 for Windows" or sometimes "Half Pack") are upgraded to OS/2 Warp Connect
 - systems with WinOS/2 support installed (a.k.a. "Full Pack") are upgraded to OS/2 Warp Connect with WinOS2

1.4.1 Installation on Workstations without Operating System

This is a workstation without any disk partition or operating system installed, which you might find with a brand new system. There is NO preloaded system software on its hard disk. These systems may also be called **pristine systems**.

The software distribution manager administrator (together with the end user) has to decide how to partition the client workstation's hard disk. There are two possibilities:

- No specific partition defined, OS/2 will install on drive C: of a partition using all available hard disk cylinders.
- Hard disk partitions are defined to separate the operating system from end user files and additional applications; this is recommended.

All of the indicated environments can be considered as no operating system workstations. In order to install OS/2 Warp Connect or OS/2 Warp Connect with WinOS2 the diskette-initiated method (*pristine installation*) will be used. This means, the client workstation will be booted with a set of **two** boot diskettes.

1.4.2 Installation/Upgrade on DOS/Windows Workstations

This is a workstation with a DOS or DOS/Windows system installed.

The software distribution manager administrator has to decide whether the hard disk of the client workstation needs to be reorganized with different partition sizes or keep the existing partitioning. In both cases there might be a need for backup and restore of data.

The software distribution manager administrator has to know where the DOS/Windows data are stored in order to run a backup procedure before starting the actual installation of OS/2.

All of the indicated environments can be considered as DOS/Windows workstations in the sense of backup, restore and/or repartitioning of the hard disk. In order to install OS/2 Warp Connect or OS/2 Warp Connect with WinOS2 the diskette-initiated method will be used. This means, the client workstation will be booted with a set of **two** boot diskettes.

1.4.3 Installation/Upgrade on OS/2 Workstations

In this scenario the complete upgrade of OS/2 operating system will be done. We will start with a brief overview of the steps needed in this process:

- Install LAN transport and redirection services
- Establish a connection to the CID server
- Install an OS/2 maintenance system using SEMAINT, boot the maintenance system and perform the upgrade

We have to distinguish several cases:

1. The current operating system has been installed using the CID method
 - a. The connection to the CID server is still active

You can start the upgrade immediately. This is an disketteless installation scenerio.
 - b. The connection to the CID server has been deactivated, but is still available on hard disk.

In this case you only have to reactivate the connection using the command line. As no files need to be installed, this is basically also an disketteless installation scenario.
2. The current operating system has not been installed using the CID method

In this case prior to upgrading the system, you have to install LAN transport and redirection services on the workstation's hard disk and activate them. The administrator should prepare a diskette containing all necessary files

1.4.4 Installation of applications

Other CID enabled applications (see Appendix J, "CID Enabled Applications" on page 583 for a list.) be installed in a disketteless scenario once the connection to the CID server has been established. The software distribution manager administrator has to define the products in the software distribution manager environment.

1.5 Overview of Workstation-Based Software Distribution Managers

This section provides brief information about IBM's current workstation-based SDMs.

1.5.1 LAN CID Utility

As far as software distribution is concerned, MPTS consists of three primary components:

- LAN Adapter and Protocol Support (LAPS)
- LAN CID Utility (LCU)
- SRVIFS

Although these components are packaged together, each component provides a separate function in the CID environment.

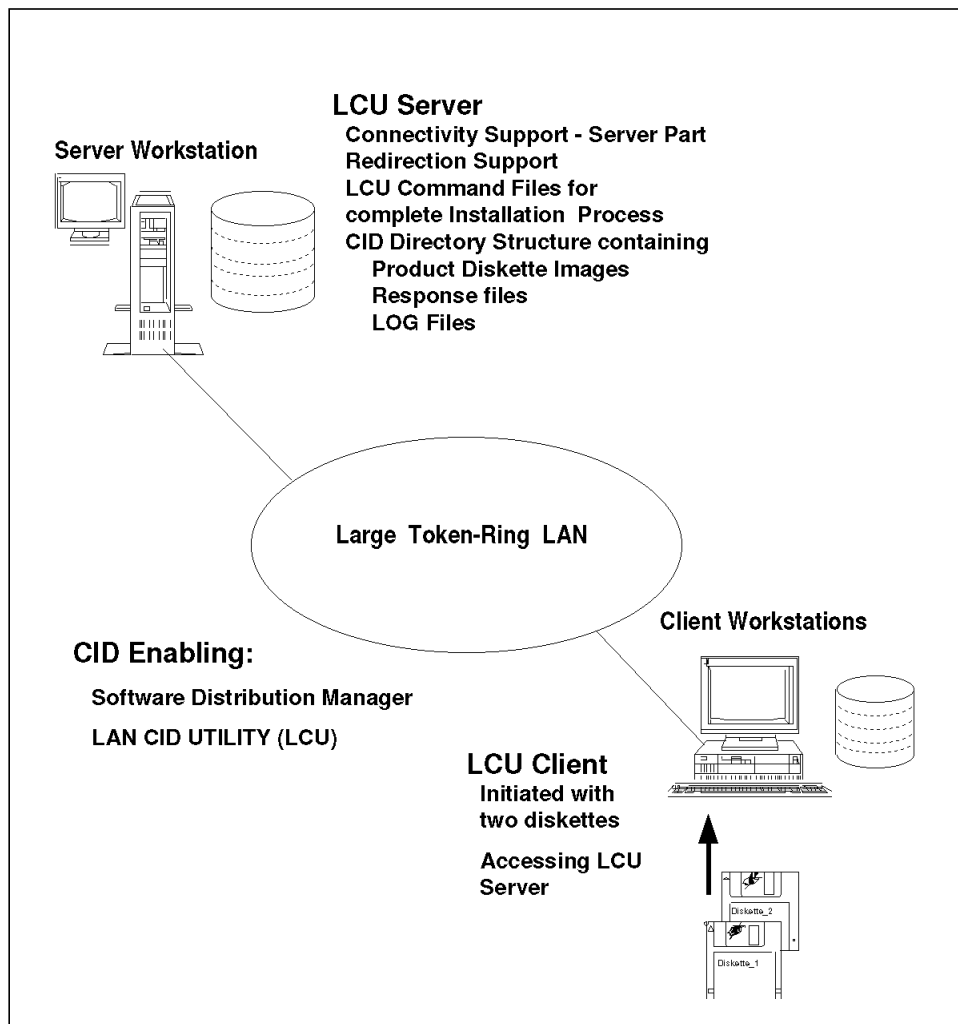


Figure 8. MPTS LCU SDM Constellation

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)

Remarks on MPTS and LAPS

A brief history of LAPS/MPTS

- 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 as a separate product is withdrawn from marketing as well as NTS/2. The successor of both products is MPTS which features more functions as well as performance enhancements.
- If you use the word *LAPS* today, you are referring to a component of MPTS.
- Today MPTS is packaged with several other products like OS/2 Warp Connect, LAN Server V4.0, LAN Server V5.0, ...

— Versions of MPTS —

Note on the current versions of MPTS

- 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 is the sole delivery vehicle for the TCP/IP protocol stack, which merges with it the socket support. Therefore, TCP/IP Version 3.0 prereqs MPTS, which supplies the protocol stack required by TCP/IP.

Thus, the two “flavors” are:

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

Why is it important to know about this difference? If you plan to update your MPTS with a CSD, 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 that comes with LAN Server V5.0
- 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. See 3.3, “Special Considerations” on page 76 for more information on how to prevent installation of a wrong MPTS level.

2. LAN CID Utility

Integrated with MPTS is the LAN 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 OS/2, MPTS, DB2/2 V2.11 and LAN Server V5.0 to be installed. Though 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 (Service Installable File System)

SRVIFS is actually a small NetBIOS based file server and requester. 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 many services required by a generalized LAN product such as LAN Server V5.0.

1.5.2 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 new *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*, SH19-6915-05, 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/without a host connection via its client/server capabilities.
- A new feature called *Remote Administrator* 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 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 at 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 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 the 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.

1.5.3 Positioning LAN CID Utility and NetView DM/2 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 and not *how* it is done. Thus, this is not intended for a comparison of implementation details.

The table summarizes the built-in features of LAN CID Utility and NetView DM/2 V2.1. It acts as a short and clear summary.

<i>Table 5 (Page 1 of 2). Positioning LAN CID Utility and NetView DM/2</i>		
Supported Features	LAN CID Utility	NetView DM/2 V2.1 (without LAN Download Utility)
Operating systems at target (client) workstations	OS/2	OS/2 DOS/Windows
Systems environments	Stand-alone LAN	Stand-alone LAN Host-connected LANs
Place of task invocation	Local workstation	Local server Remote focal point
Remote procedure invocation	no	yes

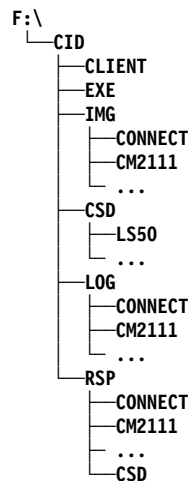
<i>Table 5 (Page 2 of 2). Positioning LAN CID Utility and NetView DM/2</i>		
Supported Features	LAN CID Utility	NetView DM/2 V2.1 (without LAN Download Utility)
Scheduled task execution	no	yes
Software Distribution Manager supporting CID-enabled installation	yes	yes
Software Distribution Manager supporting NON-CID-enabled installation	no	yes
Installation modes	Lightly attended	Unattended
Pristine client installation	Lightly attended	Lightly attended
Central tracking of resources	no	yes
ASCII/EBCDIC conversion	n/a	yes
Automatic compression before transmission	n/a	yes
Monitor status of data transmission	n/a	yes
Autorecovery after line failure	n/a	yes
Automatic disk space verification before installation (product independent)	no	no
Automatic backup before installation	no (possibility for full backup)	optional (for NON CID) no (CID-enabled installation)
Note: n/a: not applicable		

1.6 Setup of a Code Server

With this section we will introduce you to the main steps of setting up a code server. After having decided what type of software distribution manger 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.

1.6.1 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 logs etc. You will create a structure that looks like this:



1.6.2 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.

1.6.3 Manual Setup

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. And you will need a subdirectory, which will be called CLIENT, to hold the control files for a comprehensive installation process of each workstation.

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 startup diskettes to start the pristine clients with.

1.6.4 Automated Setup

Both software distribution managers supply utilities to support you in installing all products and the server/client support files onto the server in the appropriate way.

1.7 Installation Process

The installation process mainly consists of the following steps:

- Prepare the control files on the server to control which products will be installed onto the client and how they will be installed.
- Create the startup diskettes for a pristine system or create a SEED system on clients that will be upgraded.
- Start the installation process at the client, or if it is an upgrade and you have a software distribution manager like NetView DM/2 V2.1 in use, you may start the process from the server.
- Check the log files after install completion to ensure proper installation of each product.

1.8 CID Enablement of Products

CID conceptually defines six criteria for a software product to be CID-enabled. Five of these criteria address the *installation program* of the product:

- Response files
- Command line driven execution
- Redirected drives
- Progress indication and logging facilities
- Standard return codes

The sixth requirement is not directly related to the installation program:

- Transfer of product diskettes

1.8.1 Response Files

Response files provide predefined (also called "canned") responses to prompts normally aimed at the user during the installation and configuration process. They literally replace a person answering questions at installation time. Response files contain ASCII data and may therefore be created using any tool that creates ASCII output such as utilities supplied with the product (such as CMRECORD.CMD of Communications Manager/2*), an ASCII editor or REXX procedures. Since installation and configuration information may be recorded in response files, manual intervention at the time of installation is not required. A system administrator may specify this configuration information anywhere at any time. End users do not have to be involved in this process.

1.8.2 Command Line Driven Execution

Product operations such as transfer of product diskettes, installation, configuration and maintenance must have the ability to be executed from the command line with parameters that are defined by the product and/or the software distribution manager.

Command line driven execution is required because dialog-driven execution always requires manual intervention.

1.8.3 Redirected Drives

Drive redirection means installing software from a drive other than diskette drives for example A: or B:. This drive - represented by a *disk drive* letter - could be an alternative drive on the local system, a drive on a LAN or other network, or some other device that appears to the installation program as a logical drive (such as an optical device).

1.8.4 Progress Indication and 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, most 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.

1.8.5 Standard Return Codes

Return codes conforming to the CID standard need to be provided by the installation program, so the software distribution manager can check whether the installation process completed successfully or whether and which types of problems occurred. In case of an installation failure, these return codes allow remote personnel to take further measures to ensure successful installation of the product.

These return codes also provide the software distribution manager with information about required workstation reboot operations.

1.8.6 Transfer of Product Diskettes

To avoid having to exchange diskettes during the installation process, the contents of the product diskettes must be transferred to a medium that is large enough to store all of the product code. The diskettes therefore have to be copied into a directory structure on this medium (for example, a hard disk). The contents of such a subdirectory are called *diskette images*. Every CID-enabled product must have a documented method of loading its files from diskettes onto the hard disk in a directory structure understood by its installation program and the software distribution manager.

Part 2. CID System Usage and Administration

Part two is intended for the administrator of a running CID system. This is the person responsible for helping the clients by preparing the response and control files which are referenced when the client machine is generated.

Chapter 2. Recommended CID Directory Structure

This section covers the creation of the recommended CID directory structure and some considerations for LAN CID Utility (LCU) and NetView DM/2.

The purpose of the CID directory is to organize the files of CID enabled products into a standard structure on the code server.

The directory tree stores the product code images, the response files, the Service/Select Paks, and the installation log files. The recommended structure is to create individual IMG, RSP, CSD, and LOG directories for storing the product image, response files, service/select pak images and logs. To create the basic directory tree, you can use a procedure such as the MKDIR commands which you can consolidate into one REXX file to save time and keystrokes.

The images of all your products will be stored in this directory structure so it should ideally be built on a large dedicated disk partition with enough space. See Chapter 9, "Hardware and Software Requirements" on page 263 before you start with your server setup.

Many CID-enabled products have a utility that loads the files into appropriately defined directories on the code server. See Chapter 16, "Loading Product Images to Code Server" on page 379 for details on the utilities supplied by products and the disk space needed on the code server.

How to avoid problems with compression software?

Many products that are shipped on diskettes use a compression tool to limit the number of diskettes needed. You need a matching version of the counterpart of this tool to un-compress the software stored on the diskettes.

For example: If the product has been compressed by a tool called "PACK Version xyz", it is obvious, that you have to uncompress it using its counterpart software called "UNPACK Version xyz".

So, always make sure that you are using the appropriate version of the UNPACK software, which is usually shipped with the product!

To put it more precisely, if there are different versions of an UNPACK tool and all of them have the same name:

- Do not rely on the assumption that the newer versions are downward compatible
- Ensure that the search order on your system will find the version you need

2.1 CID Directory Structure Considerations

The basic directory tree for CID is shown in the following figure. Using LCU, the directory tree on the server will be accessible for a client during the installations according to the alias definitions of the server's INI file. All images, response and command files, and executables will be shared as "read only." The LOG directories are shared with "read/write" access for the client. This is done to prevent manipulations of the server by somebody while a client is connected to the server.

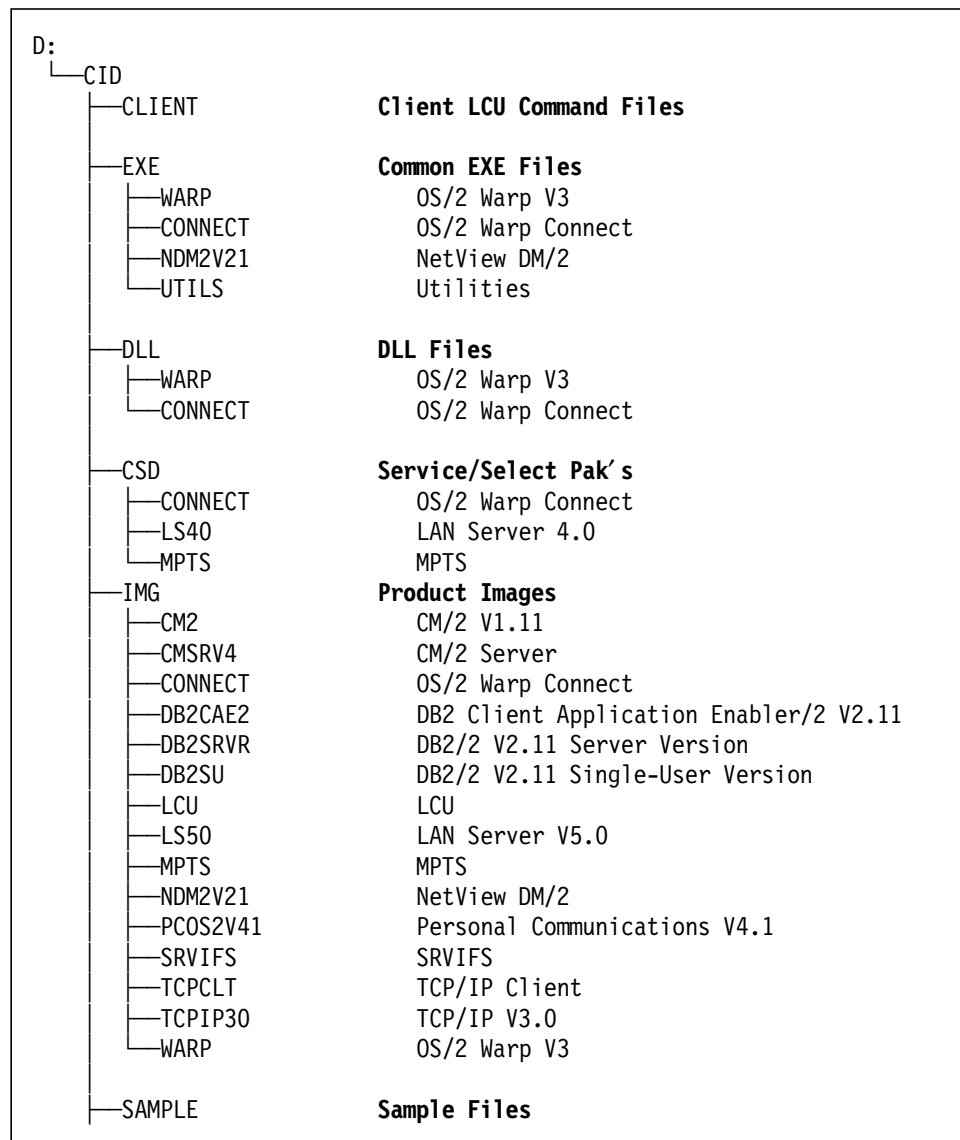


Figure 9 (Part 1 of 2). The CID Directory Structure

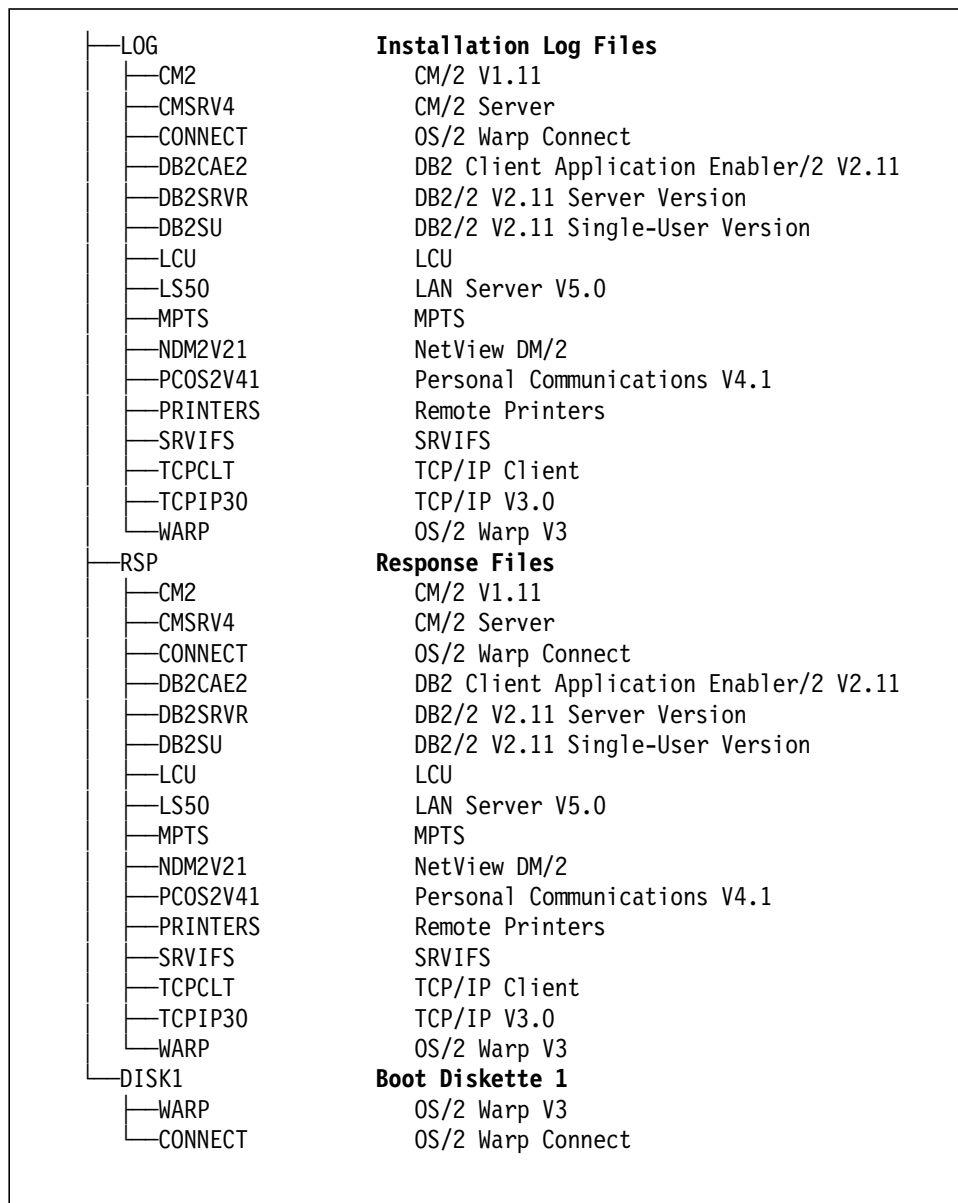


Figure 9 (Part 2 of 2). The CID Directory Structure

2.2 NetView DM/2 Directory Structure Considerations

The directory structure used with NetView DM/2 is similar to the common CID structure.

The NetView DM/2 CC server provides two shareable directories (SharedDirA and SharedDirB) that are accessible by the CC clients during an install. The directory shared via the parameter *SharedDirA* is normally shared with read access, *SharedDirB* is shared with read/write access. These parameters are defined in the configuration file for NetView DM/2, IBMNVDM2.INI, when NetView DM/2 is installed. They can easily be changed later.

Most of the publications describing NetView DM/2 usage use SHAREA as name for the directory shared with read access and SHAREB for the directory shared with read/write access. The IMG and RSP directories are therefore placed under SHAREA, the LOG directory is placed under SHAREB. Usually the log area is not under the main directory tree of SHAREA. It is not necessary to add an additional directory LOG under SHAREB. Though, this is done in most other publications and makes it easier to remember for what this directory is used.

NetView DM/2 is flexible and will support user selected CID directory structures. You can use CID instead of SHAREA and CIDLOG instead of SHAREB; just define the correct values for the SA: and SB: variables in the IBMNVDM2.INI file.

See Chapter 14, "Manual Setup of NetView Distribution Manager/2" on page 349 for detailed information on how NetView DM/2 is used for CID installs.

The DLL subdirectory is not needed, because the NetView DM/2 architecture is used to execute the installations and not the LAN CID Utility REXX utilities. The CLIENT directory is also not needed, because the LCU command files are not needed with NetView DM/2. You should create a directory to store the NetView DM/2 change file profiles that are needed to install any product. This directory can be placed anywhere, though it makes sense to put it on the same drive as the CID directory structure or directly under the IBMNVDM2 product subdirectory. We put it on the same drive as the CID directory tree.

Directory for the Profiles

MD F:\PROFILES

Assuming that your CID directory structure is on drive F:

The following figure shows the directory structure for NetView DM/2 assuming that SHAREA and SHAREB are used as root directory names instead of CID. **This is not required.** To keep this brief, only the overview of the structure with a few examples is shown. For the detailed view including all subdirectories see Figure 9 on page 41.

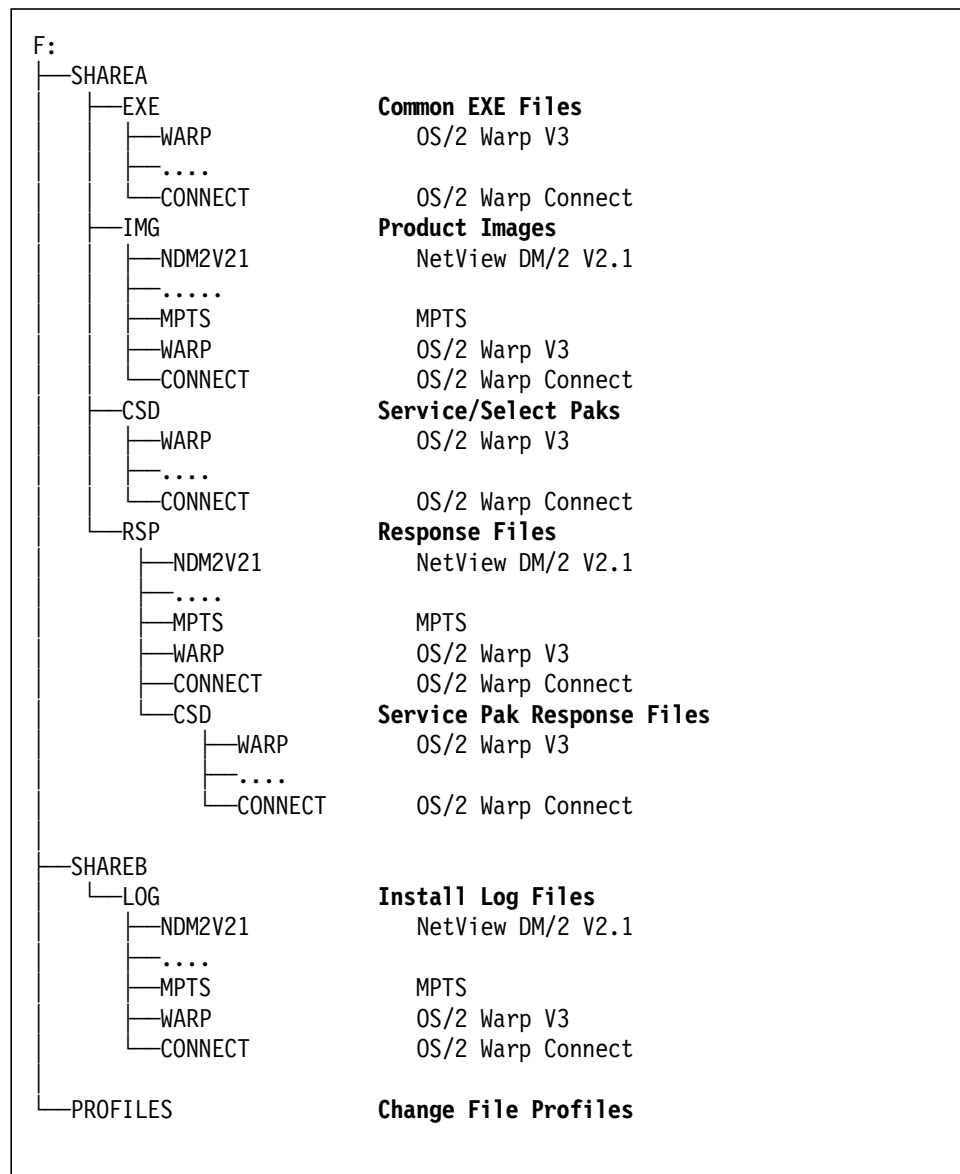


Figure 10. The NetView DM/2 Directory Structure

Chapter 3. Response Files

3.1 Introduction to Response Files

This section is an overview of response files and how these files provide the necessary configuration information required for the installation of different products.

3.1.1 Why Have a Response File?

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 with the exception of progress indicators.

In its purest form the response files used in the CID installation process replaces 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. For more information on the CID directory structure refer to Chapter 2, "Recommended CID Directory Structure" on page 39.

The response file enabled products such as Operating System/2, MPTS, Communications Manager/2, DATABASE 2 for OS/2, IBM Operating System/2 Local Area Network Server V5.0, Remote Multiple Printer Installation application, and certainly many others now and in the future, can utilize the redirected I/O and remote way of installation without user intervention.

If you are not familiar with the use of response files at all or want a better understanding on how they work please read C.4, "Response Files Basics" on page 463 in Part 5, "Appendixes."

3.2 Build Response Files

The code server administrator has to build the response files in order to install products on client workstations. This section will try to cover how to create response files for each product and point out if there is anything unique in the way the response file is interpreted by a given product.

In this section it is assumed that the code server is installed as described in Part 3, "CID System Generation and Administration." Then the sample response files, for products which supply such, are already copied to the product directories under CIDRSP.

3.2.1 OS/2 Response File

Included with each version of OS/2 there is a sample response file called SAMPLE.RSP. On an installed system SAMPLE.RSP can be found in the OS2INSTALL directory. The response file is version dependent since new keywords might have been added or slightly changed. For example there are new keywords referring to PCMCIA support in OS/2 Warp V3 and OS/2 Warp Connect that could not be found in its predecessors. Also the number of supported CD-ROM drives is usually larger in newer versions.

For details about OS/2 keywords refer to Appendix C, "OS/2 Response File Keywords" on page 433.

For OS/2 Warp V3 the response file directory is CIDRSPOS2V300. For OS/2 Warp Connect the directory is CIDRSPCONNECT. SAMPLE.RSP is copied there during the code server setup.

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. See "Default Response File" on page 150 for a description of default response file selection by LCU command files.

An OS/2 response file contains two keywords called **ExitOnError** and **RebootRequired** that are mandatory for a successful CID installation of OS/2.

The two keywords must be changed and set to the following values:

Required values

ExitOnError = 1
RebootRequired = 0

Include Files Warning

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 will install with a particular response file. Some of the more important keywords to understand are:

FormatFAT Specify the hard disk drive letters, that should be formatted using FAT file system.

FormatHPFS Specify the hard disk drive letters, that should be formatted using HPFS file system.

Support for old formatting keywords

Use the new keywords

FormatFAT
FormatHPFS

for installation of OS/2 Warp V3 or OS/2 Warp Connect.

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 SBCS) 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 until the user runs some Windows applications and are not getting the expected behavior. So ensure it is correct from the beginning!

DefaultPrinter How this parameter works is described in C.3, "Printer Description Table for AdditionalPrinters and DefaultPrinter Keywords" on page 462. It is wise to test that the choice you make really installs the printer you intended.

This is because the list of printers, PRDESC.LST, is version dependent since more printers are added for each version of OS/2.

DisplayAdapter Even though it is possible to specify 7 for SVGA support, this will not lead to a full install of SVGA as 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. Please see 4.1.1.3, "SVGA Support" on page 82.

MultimediaSupport This keyword is supported by OS/2 Warp V3 and OS/2 Warp with WinOS2 V3. If set to 1 multimedia files will be installed. But the multimedia configuration has to be done by a separate program after the OS/2 installation. For more information see 4.1.1.4, "Multimedia Support" on page 86.

In installation scenarios involving multiple partitions on one hard disk or a Boot Manager installation, please refer to 8.2, "The Fixed Disk Utility Program (FDISK)" on page 244.

3.2.1.1 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. If some client then needs some special settings you can provide a client response file, which makes an 'IncludeInLine' of the default response file and then for those keywords that differ, defines them to whatever values they should be.

For example if 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:RSPCONNECTDEFAULT.RSP  
TargetDrive=D:
```

If using NetView DM/2 V2.1 and you want to do change management it might be useful to have a complete response file for each client in order to keep track of what really is installed on the client.

Drive letters assigned

NetView DM/2 V2.1 uses the variables SA: and SB: to point to the redirected drives used during CID install.

It is not obvious which drive letters are assigned to SA: and SB: on the client. When booting from diskettes,

SA: is mapped to drive W:

SB: is mapped to drive V:

If booting from the client's hard disk,

SA: is mapped to drive X:

SB: is mapped to drive W:

If you are not sure which drive letters are used, check the MESSAGE.DAT file of the client, where you find the complete program invocation with resolved drive letters. You may want to use the specific drive letters for NetView DM/2 V2.1. For this task, you can use the response file keyword DriveLetters during installation, which is transformed to an IBMNVDM2.INI keyword. For detailed information about the DriveLetters keyword, check the NetView DM/2 V2.1 manuals.

If you want to use any of the INCLUDE keywords, you will need to know which drive letter is mapped and use this drive letter to give a fully qualified path in the response file. There is no way to pass this information to the response file. The same is needed for all other keywords that need path information (for example DDInstall, COPY and UserExit) for SA: and SB:.

Be aware that these assignments may change, if you have other products running on the client workstation, that use redirected drives.

3.2.2 Communications Manager/2

Information about CID installation of CM/2:

- can be found in directory CIDIMGCM2
 - there are three sample response files for
 - (first time) installation
 - re-installation
 - upgrade
- as well as a VIEWable file called README.INF.

For CM/2 there are keywords 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 CM/2 provides a utility **CMRECORD**, that generates Communications Manager/2 response files from an existing Communications Manager configuration.

With CMRECORD, all features of a CM/2 install 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 create a response file from default configuration type

```
CMRECORD /D
```

Additional information about CMRECORD invocation and use can also be found in the online *CM/2 Command Reference*.

Mandatory update for CMRECORD generated response file

After CMRECORD is run the resulting response file has to be edited with at least the **CMUpdateType** keyword to indicate what you want the response file to do.

You might want to add CMUserCFG and/or CMMModelCFG and/or CMStopCommunications as well.

To avoid typing errors use the tools provided to make the CM/2 response files:

1. Do a manual install using CMSETUP of the version of CM/2 you want to use on one machine.
2. Run CMSETUP on that machine in order to create model configurations:
 - a. Invoke CMSETUP.

- b. Click on **SETUP**.
- c. Change to the drive and directory where you want to store the model configuration.
- d. Enter a configuration name and a description.
- e. Answer **Yes** to the question, whether you want to create the configuration.
- f. Answer **No** to the question, whether the configuration will be used for this workstation.
- g. Answer the questions.
- h. Configure the features you want to install with this model.
- i. Verification is done after selection of **Close**.
- j. If there are no error messages, you can click on **Close** to leave the Communications Manager Setup window.
If you get error messages, do the necessary corrections now.
- k. Now your model configuration files can be found on the drive and in the directory you chose.

There should be four files with the configuration name and the following types CFG, NDF, CF2 and SEC. Take care to keep these together.

- 3. On the same machine use CMRECORD to make a response file.

The CMRECORD utility generates a Communications Manager response file from the default configuration and the installation parameters of the workstation on which it is run, or from any existing CM/2 configuration. CMRECORD ignores keylock, allowing you to create a response file from a keylocked configuration.

CMRECORD Example

```
CMRECORD C:\CMLIB\MODEL.CFG /O C:\TMP\MODEL.RSP
```

For the input file give the fully qualified path and name of your model configuration. For the output file a fully qualified path and name.

CM/2 Response File Naming

The CM/2 installation program, CMSETUP, requires that the file type of the CM/2 response file(s) is RSP.

The response file generated by CMRECORD is not complete to use with the CMSETUP /R command.

CMRECORD can be invoked with parameters so it will only create specific keywords. This is very useful when you are making client-specific response files and only want the keywords that you want to change.

4. Edit the output response file from CMRECORD and add CMUpdateType.

You may also want to add CMModelCFG.

5. Copy the response file and the model configuration files to the CIDRSPCM2 directory.

The subdirectory may be somewhere else, if you are using another version of CM/2 or another CID directory structure.

3.2.2.1 Use of Client-Specific Response File for CM/2 Installation

For CM/2 configurations 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 for CM/2 on how to use client-specific response files:

1. 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.

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 CMModelCFG keyword and provide a model configuration to be used and then provide the keywords for the client-specific settings.

The INCLUDE Keyword in CM/2 Response Files

When using an INCLUDE in a CM/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.

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

3.2.3 DATABASE 2 for OS/2 Response File

All varieties of IBM DATABASE 2 for OS/2 Version 2.11 use identical utility programs (from Software Installer) for installation and preparation of response files:

- INSTALL.EXE
- DB2RESP.EXE

In our lab we performed tests with

- DB2/2 V2.11 Single-User Version
- DB2/2 V2.11 Server Version
- DB2 Client Application Enabler/2 Version 2.11

The easiest way to create a DB2/2 V2.11 response file is to use the DB2/2 response file generation program **DB2RESP**. As mentioned above, all DB2/2 V2.11 installation related utility programs are the same. So, you can invoke DB2RESP from any of these directories:

CIDIMGDB2SU
CIDIMGDB2SRVR
CIDIMGDB2CAE2

To generate a basic response file for the *Single User* version, perform the following steps:

1. Change to the directory CIDIMGDB2SU.
2. Invoke DB2RESP.

3. The first panel is called IBM DB2 Create Installation Response File. Open the Product drop down list and select **IBM DB2 for OS/2 - Single - User**. This will update the Optional Components list below.
4. Select all the components you want to install from this list. To select all of them, you could click on one of them and then press these three keys at a time: **Control + Shift + /**.
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.

The generation of another response file for a different product set (DB2/2 V2.11 Server Version for example) is similar. The only difference is the selection of a different product in step 3). For detailed information on DB2/2 response files please refer to the documentation that comes with the product. Also make sure to check the README file for the latest updates.

3.2.3.1 Use of Client-Specific Response File for DB2/2 Installation

Except when installing DB2/2 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 DB2/2 on how to provide client-specific response files:

1. You can make one complete response file for each client.

It is rare that the DB2/2 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 SQLLIBSQLSYSTM file.

The INCLUDE Keyword in DB2/2 Response Files

When using an INCLUDE in a DB2/2 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
```

3.2.4 MPTS Response File

MPTS provides a utility called **LAPSRSP** to build response files for unattended installation of MPTS.

- LAPSRSP needs MPTS configuration files (like PROTOCOL.INI, ...) as input.
- There are two basic approaches to provide these configuration files:
 1. Use the configuration files from a running system.
 2. Build new configuration files using the MPTS installation program.
- LAPSRSP will 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 CIDRSPMPTS.
- For a detailed description of MPTS refer to *MPTS Configuration Guide*, S10H-9693

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.

Build new 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\resolve *.bak
```

Depending on your current MPTS configuration it's 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 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 LAN Server V5.0, CM/2, DB2/2 or any other application running on top of IEEE 802.2 interface later.

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.
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 2 sessions, 14 commands and 6 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:IBMCOMPOTOCOL.INI IBMCOMPOTOCOL.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:ibmcomrhc*.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
and for MPTS also
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 *

```

MPTS uses a locked files device driver and keeps track of what is supposed to be updated in OS2INSTALLIBMLANLK.LST. Check this file and remove the temporary files and directories and the IBMLANLK.LST. Otherwise MPTS will not start until these files are updated and since the intention was only to produce valid configuration files and not to update the existing system this is not desired.

Now you are ready to use LAPSRSP, which can be found in the IBMCOM subdirectory of an installed MPTS, or in the CIDIMGMPPTS directory.

LAPSRSP Syntax

LAPSRSP <source path> <target path> <options>

<source path> Fully qualified path

Specifies drive and path of source PROTOCOL.INI file.

<target path> Fully qualified path

Specifies drive and path for the resulting LAPS response file.

/T: option

Value of the MPTS response file Target parameter. This is the OS/2 drive.

/I: option

Value of the MPTS response file Install keyword. This has two values **PRODUCT** or **ADDITIONAL**. Normally **PRODUCT** is used. **ADDITIONAL** should only be used when adding something to an existing MPTS installation.

/C: option

Value of the MPTS response file CFG_PATH_NAME parameter. This is the OS/2 fully qualified filename of the OS/2 EE V1.3

Communications Manager .CFG file that will be migrated to a PROTOCOL.INI file.

/U: option

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: option

Specify fully qualified path and file name of a valid MPTS configuration file from a running MPTS installation (where it is found in the MPTNBIN directory as MPTCONFIG.INI). The MPTCONFIG.INI is used as input for the MPTS section of the response file.

/N: option

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 response file.

/B: option

Specify 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 response file.

/V: option

Specify fully qualified path and file name of a resolv list file from a running MPTS installation (where it is found in the MPTNETC directory as RESOLV).

A valid RFCBCST.LST file could look like:

```
nameserver 2.2.2.2
```

The resolv file is used within the resolv list section of response file.

The following example assumes the PROTOCOL.MOD was created as described earlier. (If you already have the LAPSRSP.RSP sample file, use another name.)

LAPSRSP example (No blanks between /option: and value!)

```
LAPSRSP C:\ibmcom\protocol.mod
      D:\cid\rsp\laps\lapsrsp.rsp
/I:product
/T:C:
/U:new
/M:C:\mptn\bin\mptcfg.mod
/N:C:\ibmcom\rfcnames.mod
/B:C:\ibmcom\rfcbcst.mod
/V:C:\mptn\etc\resolv.mod
```

3.2.4.1 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.

3.2.5 Creating Response Files for LAN Server

For detailed descriptions please see *IBM Operating System/2 Local Area Network Server V5.0 Network Administrator Reference Volume 1: Planning, Installation and Configuration*

Like all other response files for CID enabled products, the response files for LAN Server 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 Server it is recommended to use the **LANINST** program.

The following sections will take the reader through the steps for generating LAN Server response files. The first section describes how to create a requester response file and the next section describes how to create a server response file. In the examples LAN Server V5.0 version was used.

The only difference from a usual installation using LANINST is that when the response files are created there are push button choices to Use Target Setting and normally this push button has input focus. That means you have to **deselect** it to be able to enter anything into another field.

Install if Required works the same way as for the interactive LANINST installation. If the feature is required because of some other chosen feature and is not installed already or is an older version it will be installed.

For features that might be installed already, maybe by another product, it is wise to use Install if Required instead of Install, because if LANINSTR encounters an Install keyword in your response file and that feature/function already is installed with a similar or newer version LANINSTR 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.

3.2.5.1 Building a Requester Response File

1. Enter the command LANINST either from
 - the LAN Server diskette 1
 - or
 - its image, which is stored in directory CIDIMGLS50IBM500S1
2. Click on **Tailored**.
3. Select the radio button for Create a requester response file for remote installation at the Installation Tasks panel. Click on **OK**.

4. On the Response File Name window type a path and filename, where you wish to store the response file. For example:

D:\cidrsp1s501anreq.rsp

Select **OK**.

5. Ensure that you have specified the correct path/filename and select **Yes** at the pop-up confirmation window.
6. On 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. On the Hard Disk window specify the disk letter of the target workstation where the LAN Services should be installed or updated. Then select **OK**.

The Installation and Configuration panel shown in Figure 11 will appear.

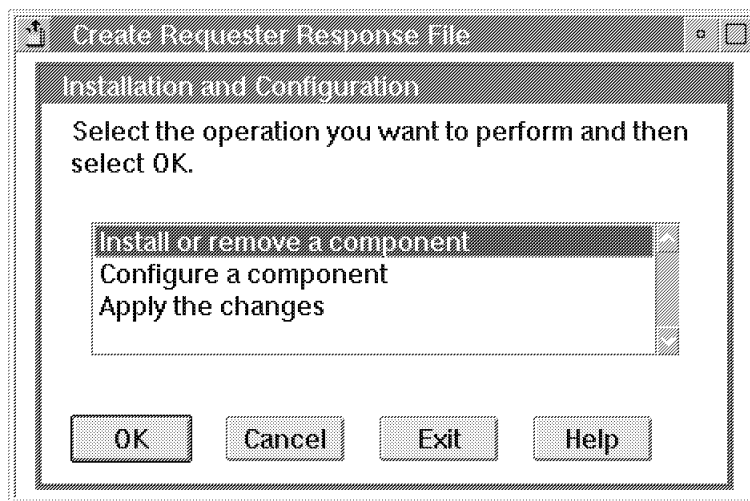


Figure 11. LAN Server V5.0 Installation and Configuration Panel

8. Select **Install or remove a component** at the Installation and Configuration panel. Then click on **OK**. The Install and Remove panel is displayed. Select the components you wish to install.

You will be given the option of Install or Install if Required.

Select **Install** if you wish to install the component selected. If there are components you wish only to install if they are prerequisites to others

you have selected, choose Install if Required. For instance, select **Install** for the component Requester, and select **Install if Required** for other components. If the target system does not have a version of LAN Server currently installed, then **Install if Required** will install all components of the Easy Install path of a standard, attended installation. If the target already has a version of LAN Server installed which is being migrated/upgraded, then **Install if Required** will install those components that existed on the previously installed system.

When you have finished selecting the components that are to be installed, select **OK**. The Installation and Configuration window is displayed again.

9. Select **Configure a component**. A list of components will be displayed which can be configured one at a time.

Component	Possible Values
Requester	Requester name, Domain name. These two parameters should be set. The requester name needs to be a unique name to ensure that the client can start working immediately after installation.
Peer Services	Share level security
LAN Services Adapters	Adapter to be used by LAN Server
First Failure Support Technology/2*	Destination of generated alerts - NetView or IBM LAN Network Manager

Select each component separately and then select **Configure**.

Notes

Any components not configured, will be migrated if currently installed on the target workstation.

If you are migrating LAN Services on a target workstation, only configure the components which will differ from the current installation. Otherwise, the configuration will override the current settings at the target.

You should always configure the requester component even if you only want to migrate LAN Services on a target workstation. This will allow you to override certain options such as auto-starting the LAN Requester.

10. In the Start Requester panel select, whether or not you wish the requester to be started automatically, when the system is IPLed.

If you are migrating LAN Services and want to use the setting previously defined on the target workstation select **Use target setting**.

Then select **OK**.

11. Now the Network Adapter - Direct Memory Access window is displayed. Some streamer type of network adapters cannot access memory above 16MB. This is because the adapters use 24-bit DMA and with a 24-bit address memory addresses above 16MB cannot be reached. When the LAN drivers load at startup they will automatically try to use memory above 16MB (if present in the system) and if the network adapter cannot handle this the system probably will hang.

If the choice At least one network adapter uses only 24-bit DMA is selected, the LAN drivers will be prevented from trying to use memory above 16MB, which also implies, that memory above 16MB will not be used for LAN Requester. Take care to make a correct choice here; if updating a system, select **Use target setting**.

12. At the Requester Services window select the services shown and set the autostart setting either on or off. Use **Use target setting**, if you are migrating the target workstation.

Peer Services will be shown, even if not selected for installation. This is normal.

Then select **OK**. The Configure window is displayed again. Select **OK**, when you are finished configuring all the components desired.

13. On the Installation and Configuration window select **Apply the changes** and select **OK**. This will cause the installation/configuration program to

build the requester response file. Select **OK** at the pop-up window, which is displayed after the creation of the response file. The Installation Task window is displayed again. You may select **Exit** here or perform another task.

3.2.5.2 Building a Server Response File

The process of creating a server response file is very similar to creating a requester response file. The primary difference will be in the number of configurable options.

1. Enter the command LANINST either from
 - the LAN Server diskette 1
 - or
 - its image, which is stored in directory CIDIMGLS50IBM500S1
2. Select **Create a server response file for remote installation** at the Advanced Server Installation/Configuration window. Then select **OK**. The Response File Name window is displayed.
3. On the Response File Name window type a path and filename where you wish to store the response file. For example:
D:\cidrsp\ls50\ansrv.rsp
Select **OK**.
4. Ensure that you have specified the correct path/filename and select **Yes** at the pop-up Confirmation window.
5. On 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. Select **OK**.
6. On the Hard Disk window specify the disk letter of the target workstation where the LAN Services should be installed or updated. Then select **OK**.
7. On the Server Type window you are prompted to select either **Additional server**, **Domain controller**, **Backup domain controller** or **Use target setting**.
8. The Installation and Configuration window, which is displayed after your selection allows you to move between the following operations:
 - Install or remove a component
 - Configure a component
 - Apply the changes

9. Select **Install or remove a component** at the Installation and Configuration window and click on **OK**. The Install and Remove window is displayed. Select the components you wish to install.

You will be given the option of Install or Install if required.

Select **Install** if you wish to install the component selected. If there are components you wish only to install if they are prerequisites to others you have selected, choose Install if required. For instance, select **Install** for the component Server, and select **Install if required** for other components.

When you have finished selecting the components that are to be installed, select **OK**. The Installation and Configuration window is displayed again.

10. Select **Configure a component**. A list of components will be displayed which can be configured one at a time (see Figure 12).

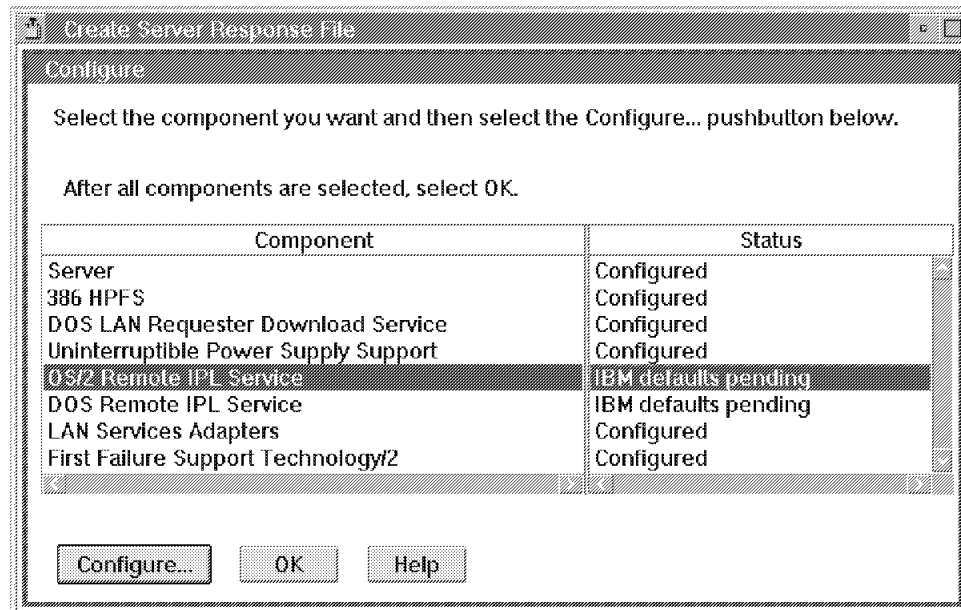


Figure 12. LAN Server V5.0 Configure Window

Notes

Any components not configured will be migrated if currently installed on the target workstation.

If you are migrating LAN Services on a target workstation, only configure the components which will differ from the current installation. Otherwise, the configuration will override the current settings at the target.

You should always configure the server component even if you only want to migrate LAN Services on a target workstation. This will allow you to override certain options such as auto-starting the LAN Server.

11. At the Start Server window select whether or not you wish the server started automatically when the system is IPLed.

If you are migrating LAN Services and want to use the setting previously defined on the target workstation select **Use target setting**.

Then select **OK**.

12. The Network Adapter - Direct Memory Access window now is displayed. Some streamer type of network adapters cannot access memory above 16MB. This is because the adapters use 24-bit DMA and with a 24-bit address memory addresses above 16MB cannot be reached. When the LAN drivers load at startup they will automatically try to use memory above 16MB (if present in the system) and if the network adapter cannot handle this the system probably will hang.

If the choice At least one network adapter uses only 24-bit DMA is selected the LAN drivers will be prevented from trying to use memory above 16MB. Which also implies that memory above 16MB will not be used for LAN Server. Take care to make a correct choice here; if updating a system select **Use target setting**.

13. At the Server Services window select the services shown and set the autostart setting either on or off. Use **Use target setting** if you are migrating the target workstation.

All available services will be shown even if not selected for installation. This is normal.

Then select **OK**. The Reinitialize Domain Controller Database window is now displayed when you choose to install the server component. In this case, if you want to preserve the DCDB that was created by a previous installation of LAN Server and currently on the target's fixed disk, select **Do not reinitialize the domain controller database**. If you want to replace

the current DCDB or are installing a new LAN Server, select **Reinitialize the domain controller database**. Select **OK**.

The Configure window is displayed again. Select the **Server** component and enter the server and domain name. Select **OK** when you are finished configuring all of the desired components.

14. On the Installation and Configuration window select **Apply the changes** and then **OK**. This will cause the installation/configuration program to build the server response file. Select **OK** at the pop-up window which is displayed after the creation of the response file.
15. The Installation Task window is displayed again. You may select **Exit** here or another task.

If you are not familiar with how to configure the components that are needed for an OS/2 LAN Server (Domain Controller, Additional Server, Remote IPL Support) please refer to *IBM Operating System/2 Local Area Network Server V5.0 Network Administrator Reference Volume 1: Planning, Installation and Configuration*.

3.2.5.3 Use of Client-Specific Response File for LAN Server Installation

The response files created with LANINST will contain all of the information required by the installation program to install a LAN Services system.

The administration of response files can certainly become a major task if the proper planning and design is not done early in the process. With the proper use of group and client response files, the task of the system administrator for maintaining response files can be minimized.

The client response file will typically start with an INCLUDE statement which will specify a group response file. The group response file will contain the keyword-value pairs that apply to a group of users.

The INCLUDE Keyword in LAN Server Response Files

LAN Server/LAN Requester response files are processed from top to bottom and duplicate keywords are overwritten when they are encountered. For this reason it does not matter whether there are duplicate keyword-value pairs. The last value specified for the keyword is used when the installation program, LANINSTR, is processing the client response file. This behavior is the same as that of the 'IncludeInLine' keyword for OS/2.

The Include keyword can only specify the response file name and the fully qualified path can not be specified. When LANINSTR searches for the include files the search order is:

- The path specified in the /G: parameter.
/G is given when LANINSTR is invoked.
- Current directory.
- Each path as set in the DPATH statement on the client's CONFIG.SYS.

Small client response files may easily be built using an ASCII editor. Assume that you have a common default response file LANREQA.RSP. In the client response file you want to change only the keywords for the client-specific information, like the requester machine name, the domain name and the workstation information used with FFST/2. In this case an example would look like:

```
INCLUDE = LANREQA.RSP
UPDATEIBMLAN = Requester<
  Computername = LRTOMAS
  Domain = ITSCBOCA
>
ConfigWsType1 = 8595
ConfigWsType2 = OKD
ConfigWsSerial1 = 23
ConfigWsSerial2 = AABR6
ConfigWsId = ITSCWK20
```

With the response file example above the requester section in IBMLAN.INI would be updated with computer name and domain, and the FFST/2 workstation information would be set.

For information regarding the supported keywords and values in LAN Server V3.0 response files, see the *IBM Operating System/2 Local Area Network*

3.2.6 NetView DM/2 V2.1 Client Feature

This section shows how to create a response file for the NetView DM/2 V2.1 client feature.

3.2.6.1 Client Response File

The client response file has two mandatory keywords:

```
CLIENTNAME=<clientname>  
SERVERNAME=<servername>
```

The SERVERNAME must match the name in the IBMNVDM2.INI file at the CC server. Additional keywords defining the log path, the timeout parameters or the drive letters may be added. If they are not specified, the defaults are used. See the *IBM NetView Distribution Manager/2 Version 2.1 Installation and Customization Guide* for detailed descriptions.

3.2.6.2 CC Server Response File

If you did not receive a softcopy of the response files, or you choose not to use them, the following shows the procedure for creating the NetView DM/2 response file. A remote install of a CC Server is not discussed in this book. You can use this response file for an install from diskettes or from the images if you want to avoid the prompting for example.

1. Open an OS/2 window and issue the following commands:

```
D:  
CD D:\SHAREA\RSP\NVDM2  
EPM NVDM2SRV.RSP
```

2. After the editor comes up, type in the lines below and then press < F 4 > to save and exit

```
//  
// Installation Parameters .....  
//  
CDM=C  
SERVER=YES  
//  
// Configuration Parameters .....  
//  
MAXREQUESTS=8  
MAXCLIENTS=20  
MAXSHRFILES=1000  
ADAPTERNUM=0
```

```
AGENTTIMEOUT=-1
MESSAGELOGFILE=C:\MESSAGE.DAT
ERRORLOGFILE=C:\ERROR.DAT
LOGOPTION=NVDM
SERVERNAME=NVDMPTEST
```

3. Note that entries for *MESSAGELOGFILE* and *ERRORLOGFILE* should be explicitly defined. Otherwise, these will default to the directory where NetView DM/2 V2.1 is installed.
4. If you want to install a CC server with focal point connection or APPC connected to other CC servers, you have to define additional parameters while others might change.

For detailed information on NetView DM/2 V2.1 keywords see *IBM NetView Distribution Manager/2 Version 2.1 Installation and Customization Guide*.

Comments

The only allowed sign for comments is the //.

3.2.7 TCP/IP Response File

A sample response file for TCP/IP V3.0 named DEFAULT.RSP can be found on the OS/2 Warp Connect CD-ROM in directory CIDIMGTCPPAPS. There is no utility available to create a response file. You should use the DEFAULT.RSP as a skeleton and adjust it to your environment. Please read carefully through the description given in the TCP/IP Command Reference, which is available as an online book (in *.INF format) on an installed system.

The syntax of the TCP/IP response file is slightly different from the syntax other products use:

- 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.
- Numerous keywords specifying the configuration can be added at the end of the response file.

Additional information can be found in Table 7 on page 124.

3.3 Special Considerations

Some features like

- MPTS
- First Failure Support Technology/2
- User Profile Management

are delivered with several products.

These features also come with new versions. A Select Pak or Service Pak for one of the products, that provides such a feature, usually tries to update the feature as well - and requires this to be done.

Normally the installation/servicing program is able to determine, whether the version installed on the system is the same or even newer level, and therefore no update should be installed.

If you have an installation program, that supports both response file parameters

- Install
and
- Install if required

choose the latter, if the feature already is installed on the system.

Some features allow you to specify an upgrade_level in the response file. In this case, select

- same
or
- old

to avoid replacing newer code by a backlevel version.

MPTS with its component LAPS (LAN Adapter and Protocol Support) is packaged with several other products like LAN Server V5.0 or OS/2 Warp Connect: A few rules to remember

- Always use the latest version of MPTS
- **Never** allow a product to downlevel your current installation
- Newer versions of MPTS
 - are always a superset of previous releases
 - provide compatibility with older versions

- For a CID install, set the upgrade_level = same (or old) in the inst_section of your response file to prevent overlaying a current release of MPTS with an older version.

First Failure Support Technology (FFST/2) is mandatory for some products and optional for others. We strongly encourage you to use FFST/2, since it is a helpful tool for problem determination.: Some products that use FFST/2:

- CM/2
- LAN Server V5.0
- DB2/2 V2.11

User Profile Management (UPM): Some products with UPM:

- CM/2
- LAN Server V5.0
- DB2/2 V2.11

Chapter 4. Client Installation Control Files

4.1 CID Installation Commands

CID-enabled products have installation programs that enable the products to be installed from a redirected drive. The installation programs also read information from response files. Sometimes it is the same installation program as used when the user is interactively installing the product and is answering the questions. Other products have special installation programs for CID installation.

In this section the CID installation programs are described. They are product dependent and are invoked in the same way no matter what software distribution manager you are using.

The CID installation commands are installed on the code server when loading the product's diskette images to the code server. How you do this will be explained in Chapter 16, "Loading Product Images to Code Server" on page 379.

4.1.1 OS/2

OS/2 CID utilities are shipped with OS/2. The CID-Utilities come together with OS/2 Warp Connect and are packed on disk no. 7 in the CID-Bundle.

4.1.1.1 SEINST

SEINST.EXE is intended to install OS/2 on the client workstations. SEINST is normally run under a maintenance system created by SEMAINT or on boot diskettes created by SEDISK.

SEINST calls the RSPINST.EXE, which is the OS/2 installation executable. RSPINST.EXE is described in 4.1.1.2, "RSPINST" on page 82. SEINST accepts parameters in a different format and will perform some cleanup related to SEMAINT.

SEINST relies on an environment variable called **REMOTE_INSTALL_STATE**. This variable will be discussed more in 4.4, "LCU Command File" on page 143.

If **REMOTE_INSTALL_STATE=0**, SEINST will restore the CONFIG.SYS, AUTOEXEC.BAT, and STARTUP.CMD of the workstation (which were saved as CONFIG.S13, AUTOEXEC.S13, and STARTUP.S13 by SEMAINT) before calling RSPINST.

SEINST Syntax

```
SEINST /S:<source path> /B:<target drive> /T:<boot directory>  
/R:<path><response file> /L1:<path><log file name>
```

/S: Fully qualified path to the OS/2 diskette images

This can be a local hard drive or redirected drive.

/B: Target boot drive

This is the drive OS/2 will be installed to. The system will be booted from this drive **after** the installation.

The drive specified **must** be a local drive.

/T: Directory from which the client is booted during the installation

This is the directory which SEMAINT installed the OS/2 maintenance system to and from which OS/2 is booted when SEINST is invoked. In this case the parameter is required. Or if the boot drive is removable (for example A:), then this parameter is optional. If the parameter is specified anyway, then the value is not verified.

If it is a maintenance directory SEINST will delete all files and remove the directory after the installation, so if it is anything that should be kept it has to be copied before SEINST is invoked.

/R: Fully qualified path and name of a response file

/L1: Fully qualified path and log file name

The directory must already exist.

SEINST Example

```
SEINST /S:X:\img\CONNECT  
/B:C:  
/T:C:\service  
/R:X:\rsp\CONNECT\CONNECT.rsp  
/L1:L:\CONNECT\log.log
```

The command above will install OS/2 Warp Connect in to the C: drive, having booted from the C:SERVICE directory. The response file to be used is CONNECT.RSP from redirected X:RSPCONNECT directory, and the source path for diskette images is the directory X:IMGCONNECT. A log file will be written to L:CONNECTLOG.LOG, and the C:SERVICE directory will be cleaned up and removed.

```
x.seinst300 = 2                                /* structure index */
x.2.name=' OS/2 Warp Connect'                  /* product name */
x.2.statevar = 'CAS_' || x.2.name               /* state variable name */
x.2.instprog = 'x:\exe\CONNECT\seinst'          /* install program */
          ' /b:' || bootdrive || '            /* - bootdrive */
          ' /s:x:\img\CONNECT                  /* - source directory */
          ' /t:c:\service                      /* - service directory */
          ' /l:l:\CONNECT\' || client || '.log' /* - log file */
          ' /r:'                               /* - response file flag */
x.2.rspdir = 'x:\rsp\CONNECT'                  /* response file dir. */
x.2.default = 'default.rsp'                   /* default rsp file */

where x: is the shared drive to the code server's subdirectory \CID
      Access to x: is usually defined as 'Read only'
and y: is the shared drive to the code server's subdirectory \CID\LOG
      Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)
```

Figure 13. Extract of an LCU Client Command File Illustrating SEINST Program Invocation

OS/2 Warp V3 includes a utility for a selective uninstall. This uninstall can be invoked through the icon in the system folder or from command line. The user is then guided through menus comparable to the installation menus. The executable is called **UNINSTAL** and can be found in the OS2INSTALL subdirectory. In this subdirectory there is also an UNINSTAL.RSP. It is possible to invoke UNINSTAL with the input of the response file by entering:

UNINSTAL uninstal.rsp

The deleting of files as defined in the uninstal.rsp takes place, without any menu-driven user input. At the end of the processing, a pop-up window is displayed that has to be quit by clicking on the **OK** button. The uninstall of OS/2 Warp V3 is currently not CID-enabled.

4.1.1.2 RSPINST

RSPINST is an OS/2 executable, and allows the OS/2 installation from a redirected drive. RSPINST is normally called by SEINST, so you should not use RSPINST yourself. This EXE file recognizes the subdirectory structure created by SEIMAGE utility as described in 16.1.1, "Loading OS/2 Diskette Images with SEIMAGE" on page 379. RSPINST.EXE also interprets an ASCII response file containing all definition keywords instead of prompting the user via dialogs. The OS/2 response file keywords are described in Appendix C, "OS/2 Response File Keywords" on page 433.

RSPINST accepts a single parameter, which is the name of the response file to be used as shown below.

RSPINST Syntax

```
RSPINST <response file>
```

<response file> Fully qualified path and name of an OS/2 response file

RSPINST Example

```
RSPINST X:\rsp\CONNECT\sample.rsp
```

4.1.1.3 SVGA Support

In OS/2 V2.11 the installation of SVGA adapters is not CID-enabled at all. During the installation of the base Operating System it is only recognized that there is an SVGA adapter in the system. The installation of the drivers and setting of the refresh-rate have to be done separately. For example in the PS/2 Systems with an S3-Chipset, the driver diskettes are delivered with the system. First you have to install the DOS-Support that is used to create the SVGADATA.PMI file. The setting of the refresh-rate is done with this DOS-program too. After this DOS-support is installed, the S3-Adapter drivers can be installed manually using DSPINSTL.

In OS/2 Warp Connect the SVGA support is installed during the installation of the base-system. The lowest resolution is taken by default and it is not configurable to switch directly to a higher resolution after the first boot. To change the resolution **DSPINSTL** can be used. DSPINSTL is the OS/2 utility to install display drivers. It can be found in the OS2INSTALL directory.

In OS/2 Warp V3 it is invoked with the following parameters:

/PD:	Fully qualified path Specifies drive and path to the file where the information about the possible resolutions are stored. These *.DSC files normally reside in the OS2INSTALL subdirectory.
/T:	Fully qualified path Specifies the boot drive.
/S:	Fully qualified path Specifies drive and path to the drivers. This will be the server's IMGCONNECT directory in most cases.
/RES:	Resolution Value of the desired resolution that will take effect after the next reboot. It is to be entered in the following format: 1024x768x256
/U	installation manner Identifies the unattended install.
/AUTO	Auto-detection mode Specifies that the best .DSC file match for the installed video adapter will be determined and used automatically. The /PD: parameter is not needed when /AUTO is used.

DSPINSTL Invocation Example

```
DSPINSTL /PD:c:\OS2\INSTALL\pss3.dsc /S:X:\img\connect
/T:c: /RES:1024x768x256 /U
```

To find out which .DSC file is needed, the following table is provided. You have to know which SVGA adapter you are using. To identify which SVGA adapter you have in a PS/ValuePoint, refer to Chapter 4, Video Subsystem, in *&vpbook.*, &vpbooki.. The following figure shows the manufacturer codes that are related to the OS/2 Warp V3 supported SVGA adapters.

Table 6 (Page 1 of 2). Overview of SVGA Adapters and Their Corresponding .DSC Files.

The resolutions supported by each driver depend on the amount of video memory and on the resolutions supported by the display driver.

.DSC file name	Adapter type	Chip type
PSHEAD.DSC	VIDEO7	HT205 HT208 HT209
PSTRID.DSC	TRIDENT	TR8800 TR8900
PSTSENG.DSC	TSENG	ET3000 ET4000
TLIW32.DSC	TSENG	ET4000W32 ET4000W32IREVA ET4000W32IREVB ET4000W32IREVC ET4000W32IREVD ET4000W32PREVA ET4000W32PREVB ET4000W32PREVC ET4000W32PREVD
PSWD.DSC	WESTERN DIGITAL	PVGA1A PVGAB PVGA1C PVGA1D WD90C26 WD90C27
PSWDC31.DSC	WESTERN DIGITAL	WD90C31
PSWDC24.DSC	WESTERN DIGITAL	WD90C24
WDC33.DSC	WESTERN DIGITAL	WD90C33
PSATI.DSC	ATI	ATI18800 ATI28800
ATIM32.DSC	ATI	ATI68800MACH32
ATIM64.DSC	ATI	ATI88800MACH64

Table 6 (Page 2 of 2). Overview of SVGA Adapters and Their Corresponding .DSC Files.

The resolutions supported by each driver depend on the amount of video memory and on the resolutions supported by the display driver.

.DSC file name	Adapter type	Chip type
PSSPDW.DSC	IBM	IBMSVGA
PSC1.DSC	CIRRUS	GD5420 GD5422 GD5424
C154X.DSC	CIRRUS	GD5426 GD5428 GD5429 GD543X GD5434
PSS3.DSC	S3	S386C80X S386C928
S3864.DSC S38641M.DSC	S3	S386C864 S386C964
WP9000.DSC	WEITEK	P9000
WP9100.DSC	WEITEK	W5186 W5286 P9100

DSPINSTL can be executed as a post-installation program, after the initial install of OS/2 Warp V3 is done and the system is restarted. This is because it requires that PM is available. The sample LCU command files provided on the sample code CDROM include a product definition part for DSPINSTL. This definition has to be changed to reflect the proper *.DSC file.

Note

The refresh-rate is not set to the highest possible value with the **DSPINSTL** Program. It has to be set in the System-Object in the System-Configuration Folder.

4.1.1.4 Multimedia Support

The multimedia part that comes with OS/2 is named Multimedia Presentation Manager/2 (MMPM/2). In OS/2 V2.x it has to be installed separately after the initial install is done. The MMPM/2 install of OS/2 V2.x is not CID-enabled.

The multimedia install in OS/2 Warp V3 is CID-enabled. It is divided into two steps:

1. During the initial install the necessary files for multimedia are copied if this is specified in the response file by the keyword:
MultimediaSupport=1
2. After the system is restarted and has PM available, the configuration of the multimedia part has to be done by invoking **MINSTALL**. MINSTALL can be invoked with parameters supporting the CID installation:

Note

MINSTALL must be run from the MMTEMP subdirectory.

```
MINSTALL /M /C:<file name>
```

Where <file name> is the fully qualified path to a file that captures the configuration. Using this command, you can create a response file for further installs. The multimedia devices that are configured during that install do not have to be installed in the machine where this command is executed.

After a response file is created with the /C: parameter, it can be used for the next installs by executing:

```
MINSTALL /M /R:<file name> /L:<log file name>
```

The /M parameter tells MINSTALL to transfer files from the MMTEMP directory to where they need to be for MMPM/2 to run.

The /R parameter is the fully qualified filename of a response file created earlier when MINSTALL was run with /C.

The /L parameter is a fully qualified file name for the MINSTALL log file.

MINSTALL can be executed as a post-installation program, after the initial install of OS/2 Warp V3 is done and the system is restarted with a fully functional PM environment. The sample LCU command files provided on the sample code CDROM include a product definition part for MINSTALL.

4.1.1.5 SEMAINT

SEMAINT.EXE is intended to install a minimal OS/2 system on the target client workstation's hard disk. This minimal OS/2, also known as a maintenance system, will not contain the Presentation Manager or Workplace Shell features of the normal OS/2. By booting from the maintenance system, the normal system files will not be locked, allowing new systems (SEINST) or Service Pak (FSERVICE) installations. (SEINST and FSERVICE can also be run if booted on diskettes created with SEDISK.)

SEMAINT saves the existing CONFIG.SYS, AUTOEXEC.BAT and STARTUP.CMD to the service subdirectory as CONFIG.S13, AUTOEXEC.S13 and STARTUP.S13.

SEMAINT Syntax

```
SEMAINT /S:<source path> /S2:<service pak> /T:<target path>  
        /B:<boot drive> /L1:<path> <log file name>  
        /P:<pcmcia_id#>
```

/S: Fully qualified source path to the OS/2 diskette images

This can be a local hard drive or redirected drive.

/S2: Fully qualified source path to the OS/2 Service Pak diskette images

This parameter is used to apply the OS/2 Service Pak to the maintenance system being created.

This parameter is optional, but should be used when:

- Installing an OS/2 Service Pak
- Installing a non-OS/2 Service Pak, such as LAN Server, on a client that already has an OS/2 Service Pak applied.

When it is used care should be taken to point to the same version of the Service Pak as was **previously** applied to the client. This is especially important when applying a non-OS/2 Service Pak. If the /S2: is not supplied or is pointing to the wrong version of OS/2 Service Pak there is a risk of the OS2KRNL and OS2LDR being at the wrong level when the system is returned to the normal environment.

It should **not** be applied to the maintenance system when:

- Migrating to a new base OS/2 release.

- Installing a non-OS/2 Service Pak on a client that has OS/2 at a base level (that is no OS/2 Service Pak applied).

/T: Fully qualified target path

The maintenance OS/2 system will be installed under this directory.

The target directory must be on a local drive. If the target directory is on an HPFS drive, then the boot drive (/B:) must also be an HPFS drive.

/B: Target boot drive

The drive from which the OS/2 maintenance system will boot. The drive specified must be a local drive.

/L1: Fully qualified path and name of the log file

The directory must already exist.

/P: Optional parameter for PCMCIA support (only valid for OS/2 Version 3 systems)

This is an optional parameter when pcmcia driver support is needed. When the /P: option is used, the PCMCIA.SYS driver (as well as the appropriate socket driver) will be copied over to the boot drive. The pcmcia_id# represents a number associated with the computer desired. Look at the default response file at the keyword PCMCIA to figure out what number to put in here. For example /P:1 would be used if you need to boot on an AMBRA486 SN425C. **pcmcia_id#** must be a number representing a valid parm of keyword PCMCIA in the default response file. **pcmcia_id#** can not be 0.

OS/2 Warp Connect SEMAINT Example

```
SEMAINT /S:X:\img\CONNECT
        /T:C:\service /B:C:
        /L1:L:\CONNECT\log.log
```

The command above will install a bootable OS/2 Warp Connect maintenance system without any Service Pak in the C:SERVICE directory, using OS/2 images from the redirected X:IMGCONNECT directory. It will also write a log file to redirected L:CONNECTLOG.LOG.

```

x.semaint = 4                                /*structure index */
x.4.name=' OS/2 WARP MAINTENANCE system'      /* product name */
x.4.statevar = 'CAS_' || x.4.name             /* state variable name */
x.4.instprog = 'x:\exe\CONNECT\semaint        /* install program */
               ' /s:x:\img\CONNECT           /* - source directory */
               ' /s2:x:\csd\CONNECT\xr_W017  /* - csd directory */
               ' /t:c:\service               /* - target directory */
               ' /b:' || bootdrive || '      /* - target boot drive */
               ' /l1:L:\CONNECT\' || client || '.log' /* - log file*/
x.4.rspdir = ''                               /* no auto selection */
x.4.default = ''

where x: is the shared drive to the code server's subdirectory \CID
      Access to x: is usually defined as 'Read only'
and y: is the shared drive to the code server's subdirectory \CID\LOG
      Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 14. Extract of an LCU Client Command File Illustrating SEMAINT Program Invocation

OS/2 Warp Connect SEMAINT Example

```

SEMAINT /S:X:\img\CONNECT
        /S2:X:\csd\CONNECT\xr_W017
        /T:C:\service /B:C:
        /L1:L:\CSD\CONNECT\log.log

```

4.1.2 LAN Adapter and Protocol Support

4.1.2.1 NTS/2 LAPS

As MPTS is the successor of NTS/2 LAPS the Installation of this Product is described in the following chapter. The Installation-Program Parameters haven't changed so you can use the LAPS.EXE Program with the same parameters as described for the MPTS-Installation Program.

4.1.2.2 MPTS

This utility creates a LAN transport system. MPTS is the successor of NTS/2 LAPS used to install and configure LAN Adapter and Protocol Support. (LAN Adapter and Protocol Support is sometimes abbreviated LAPS, so it is not always the installation/configuration program that is talked about, which can cause some confusion.)

The same installation program, MPTS, is used (without parameters) for user interactive installation and configuration. Usually, since it is a PM program,

it requires a fully functional OS/2 system to run in. As you see below the correct invocation of the /E parameter is needed when running in an OS/2 CID environment without PM. MPTS interprets a response file, copies necessary LAN transport files from the diskettes, and updates CONFIG.SYS, LIBPATH, DPATH and DEVICE statements. (MPTS LAN Adapter and Protocol Support is on the MPTS diskettes 1 and 2.)

MPTS Syntax

MPTS /E: /S: /T: /TU: /R: /G: /L1:

/E: OS/2 executing environment for MPTS Installation

This parameter is optional, but you need to use it when doing a response file driven MPTS installation using a software distribution manager. This is because the default is PROD and as you can see below this requires a fully functional OS/2 system to be running.

Valid choices are either PREP, MAINT or PROD.

PREP

Used to prepare a CID hard disk-initiated system installation. When used LAPS does not create an IBMLVL.INI file, thereby hiding the LAN transport. This requires that a fully functional OS/2 system, including PM, is booted when the MPTS command is run.

MAINT

Executing environment is an OS/2 "seed" system, where PM is not available. MPTS uses a special OS/2 DLL to build the IBMLVL.INI file. This DLL will not run in a regular OS/2 system.

PROD

Executing environment is a fully functional OS/2 production system. This is the default value.

/S: Fully qualified source path

If response file is used this parameter is required.

/T: Fully qualified target path

This parameter is optional.

Default value is current boot drive. The value of the **Target**

keyword in a LAPS response file overrides the value of /T: parameter.

Incorrect value of this parameter will result in program termination.

/TU: Fully qualified path of CONFIG.SYS file to be updated

This parameter is optional.

The value of this parameter defines the location of CONFIG.SYS file to update. Default is the value specified by /T:.

/R: Fully qualified path and name of a MPTS response file

If response file is used this parameter is required.

/G: Fully qualified path of a general MPTS response file

Specifies the drive and path of response files that are **included** in the response file indicated with the /R: parameter.

If incorrect value is given MPTS will not find location of a general MPTS response file and fail.

This parameter is optional. If not specified MPTS searches the current DPATH to find the included response file(s).

/L1: MPTS log file name

This parameter is optional.

This parameter is the fully qualified path and file name of the MPTS log file, LAPSHIST.LOG, which will be copied to the code server.

MPTS Example

```
MPTS /E:MAINT /S:X:\img\MPTS
      /T:C:\ /TU:C:\
      /R:X:\rsp\MPTS\MPTS.RSP
      /L1:L:\MPTS\MPTS.log
```

The parameter for OS/2 maintenance environment is defined. The command above will install MPTS from source directory X:IMG\MPTS to target drive C:. CONFIG.SYS on drive C: will be updated with DEVICE, LIBPATH and DPATH statements. MPTS.RSP will be interpreted by MPTS. The log file is L:MPTSMPTS.LOG.

```

x.mpts = 5                                /* structure index */
x.5.name='MPTS Installation (no PM)'       /* product name */
x.5.statevar = 'CAS_' || x.5.name          /* state variable name */
x.5.instprog = 'x:\img\MPTS\mpts           /* install program */
              ' /e:maint                  /* - installed from ... */
              ' /s:x:\img\MPTS            /* - source directory */
              ' /t:' || bootdrive || '\    /* - target directory */
              ' /l:l:\MPTS\ ||client || '.log /* - log file */
              ' /r:'                      /* - response file flag */
x.5.rspdir  = 'x:\rsp\MPTS'                /* response file dir. */
x.5.default = 'MPTS.rsp'                  /* default rsp file */

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
and l: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 15. Extract of an LCU Client Command File Illustrating MPTS Program Invocation

4.1.2.3 THINLAPS

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 "seed" system is updated with DEVICE and RUN statements. The following restrictions apply:

Only adapter network drivers shipped with MPTS are supported. If you want to use another adapter please follow the instructions in Appendix E, "LAN Network Adapters" on page 489.

OS/2 V1.3, OS/2 V2.x, OS/2 Warp V3 and OS/2 Warp Connect are supported.

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. 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. (This is not a problem if you have Micro Channel* machines.)

Note

If you have added a new network adapter it might be necessary to provide a customized PROTOCOL.INI file by using the /P parameter when invoking THINLAPS.

The following files are installed on the target by THINLAPS:

LANMSGDD.OS2

PROTMAN.OS2

NETBEUI.OS2

NETBIOS.OS2

LT0.MSG

PRO.MSG

NETBIND.MSG

LANMSGEX.EXE

MAC driver

All copyfiles indicated in the *NIF file name* parameter.

THINLAPS Syntax

THINLAPS <source path> <target> <NIF file name> /P:

<source path> Fully qualified source path

<target> Target drive

Target drive for seed system. Accepted value is drive letter with a colon followed by a backslash.

<NIF file name> Network_Adapter_Driver_NIF_Filename

The name of the .NIF file that corresponds to the network adapter driver that will be stored on the target. See Appendix E, "LAN Network Adapters" on page 489 for mapping of the LAN Transport network adapter drivers and the associated .NIF file names.

/P: Fully qualified path and name of PROTOCOL.INI

The value supplied is the fully qualified file name of a PROTOCOL.INI 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.

THINLAPS Example

```
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:\CIDIMGLAPS using IBMTOK.NIF.

Note: If the client machine is an ISA-bus machine and you are using **ibmtok.nif** you 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.

4.1.2.4 LAPSDEL

This utility deletes the seed LAN transport system generated by CID utility THINLAPS. There are no parameters specified for LAPSDEL.

LAPSDEL Syntax

```
LAPSDEL
```

4.1.3 LAN CID Utility

Not all the utilities will be presented here, some are discussed in the context which they are used. As this chapter deals with CID installation programs, those that are called from LCU command files to install various parts of LCU clients are presented below.

4.1.3.1 THINIFS

This utility will place all necessary SRVIFS redirection files on the target hard disk or LAN transport diskette.

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. And generate a SRVATTCH statement in the client's CONFIG.SYS. When it is executed using SRVIFS the client connects to the desired server and gets the desired redirected drive.

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

In the sample LCU command file provided with this book THINIFS is called twice. The first time is to create a SRVATTCH statement that will connect to the code server and get 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 SRVATTCH 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 CIDLOG directory with subdirectories.)

THINIFS Syntax

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

/S: Fully qualified source path

Value supplied is the fully qualified path to the SRVIFS code on the code server (usually in IMGSRVIFS).

If omitted or if an illegal value is given THINIFS will terminate.

/T: Fully qualified target path

The target 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. MPTS THINIFS also supports long directory names.

THINIFS will terminate if an unsupported drive is supplied.

/SRV:

Name of SRVIFS code server

NETBIOS name of the code server that is used in the **SRVATTCH** statement.

Supported Values

nnnnnnnn

With NTS/2 THINIFS 1-8 *alphanumeric* characters are supported for the server name.

With MPTS THINIFS 1-15 *alphanumeric* characters are supported 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 .INI file.

Path = D:CID

In the SRVIFS server's .INI file the client will be attached to the 'root' of the CID directory structure.

***P (prompts for the server name)**

This value gives a prompt where the server name can be entered.

servernamealias

This value should equate to the *servername* 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:

Name of SRVIFS client

Value supplied is the NETBIOS name of the client in the IFS statement of the client's CONFIG.SYS file.

Supported Values

nnnnnnnn

With NTS/2 THINIFS 1-8 *alphanumeric* characters are supported for the client name.

With MPTS THINIFS 1-15 *alphanumeric* characters are supported for the client name.

Even though the SRVIFS server and client names are NetBIOS names, SRVIFS does not follow the NetBIOS naming convention.

*** (wildcard)**

This value will randomly generate a client NetBIOS name.

***P (prompts for the client name)**

This value gives a prompt where the client name can be entered.

— MPTS THINIFS —

If you want the user to have a customized prompt for the requester 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 LAN Server V4.0/LAN Server V5.0 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: SRVATTCH redirected drive

Value supplied is one alphabetic character to be used as the drive letter on the SRVATTCH statement.

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 SRVATTCH statement already exists and the drive letter is used.

THINIFS will terminate if value is not supplied.

- /TU:** **Fully qualified path of CONFIG.SYS**
- This parameter is optional.
- Value supplied is the fully qualified path of the CONFIG.SYS that will be updated.
- /NS:** **Option for IFS statement**
- This parameter is optional.
- 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 :P.**This parameter is optional.
- 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.
- Value is a **0** or **1** (default is 0).
- /W** **Option for IFS statement**
- This parameter is optional.
- 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.

THINIFS Example

```
THINIFS /S:X:\img\srvifs /T:A:
      /SRV:cidsrv
      /REQ:*P
      /D:X /TU:A:\
      /NS:5 /A:0 /W
      /L1:L:\srvifs\srvifs.log
```

The user at the client workstation will be prompted for the client name and then connected to server "cidsrv".

```

x.thinifs1 = 19                                /* structure index */
x.19.name='SRVIFS Requester 1'                 /* product name */
x.19.statevar = ''                             /* state variable name */
x.19.instprog = 'x:\img\srvinfos\thinifs      ', /* install program */
               ' /s:x:\img\srvinfos          ', /* - source directory */
               ' /t:' || bootdrive || '\srvinfosrq', /* - target directory */
               ' /tu:' || bootdrive || '\      ', /* - config.sys locat */
               ' /l:l:\srvinfos\' || client || '.log', /* - log file */
               ' /req:' || client || '        ', /* - requester name */
               ' /srv:CIDSRV                  ', /* - server name */
               ' /d:x'                        ', /* - remote drive */
x.19.rspdir = ''                               /* no auto selection */
x.19.default = ''

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
and l: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'
       (see Figure 9 on page 41)
and client is a variable containing the NETBIOS name of the client currently executing
       the command file.

```

Figure 16. Extract of an LCU Client Command File Illustrating THINIFS Program Invocation

4.1.3.2 IFSDEL

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. All return information is provided by the return codes; see K.3, "IFSDEL CID Return Codes" on page 601.

IFSDEL Syntax

IFSDEL /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.

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.

IFSDEL 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 Option

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. Authlist 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.

IFSDEL does not remove itself from the system.

IFSDEL Example

```
IFSDEL /T:C:\ /TU:C:\
```

```

x.ifsdel = 21                                /* structure index */
x.21.name='SRVIFS Cleanup'                   /* product name */
x.21.statevar = ''                           /* state variable name */
x.21.instprog = 'x:\img\srvifs\ifsdel',       /* install program */
               '/t:' || bootdrive || '\srvifsrq', /* - target directory */
               '/tu:' || bootdrive            /* - config.sys locat. */
x.21.rspdir = ''                             /* no auto selection */
x.21.default = ''

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
and y: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 17. Extract of an LCU Client Command File Illustrating IFSDEL Program Invocation

4.1.3.3 CASINSTL

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.

NTS/2 CASINSTL Syntax

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

MPTS 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 *ULLA*, LCU will try to find and execute an *ULLA.CMD* in the directory path defined with this parameter.

/D Option

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 LCU.CMD will log to /L2 and CASINSTL will not log at all.

For MPTS CASINSTL 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:IMGLCU* (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: Option

Specifies the values of LIBPATH and DPATH statements to be added to LCU redirector's CONFIG.SYS.

/O Option

Indicates that this is the first time the CASAGENT program has been called.

MPTS CASINSTL supports some additional parameters:

/D: Option

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 filename 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 path is DISK1OS2Vxx if you used the recommended directory structure.

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 CID 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 prepping 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
```

Warning: 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 System Diskette Example

```
CASINSTL /TU:A: /CMD:X:\CLIENT /D
/PA:X:\IMG\LCU
/PL:X:\DLL\CONNECT;X:\IMG\LCU
/L1:L:\lcu\log1.log
/O
```

```

x.casinstl = 22                                /* structure index */
x.22.name='LCU'                                /* product name */
x.22.statevar = ''                             /* state variable name */
x.22.instprog = 'x:\img\lcu\casinstl',          /* program name */
               '/cmd:x:\EXE\CONNECT',          /* location of .cmd files */
               '/tu:' || bootdrive || ',       /* config.sys location */
               '/pl:x:\dll;x:\img\lcu;',        /* string to add to libpath */
               '/pa:x:\img\lcu',               /* path to LCU code on srvr */
               '/l1:L:\lcu\' || client || '.log ', /*CASINSTL log file*/
               '/l2:L:\lcu\SRVIFS_REQ.log',     /* CASAGENT log file */
               '/D:CONNECT.CMD'
x.22.rspdir = ''                               /* no auto selection */
x.22.default = ''

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
and l: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 18. Extract of an LCU Client Command File Illustrating CASINSTL Program Invocation

4.1.3.4 CASAGENT

CASINSTL creates a STARTUP.CMD file on the client's boot drive. In STARTUP.CMD CASAGENT is called with parameters and this depends on the parameter values given when invoking CASINSTL.

You should use CASINSTL because it also updates the client's CONFIG.SYS. The information below is merely so that you can check that you got the expected result after running CASINSTL.

NTS/2 CASAGENT Syntax

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

MPTS CASAGENT Syntax

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

/CMD: Fully qualified path

Fully qualified path of the redirected drive that contains LCU command files.

/D **Optional**

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: **Optional**

Only supported by MPTS 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.

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 Example** ——

```
CASAGENT /CMD:X:\CLIENT /D:CONNECT.CMD
/L1:L:\lcu\CLIENT.log
```

4.1.3.5 CASCKREX

MPTS CASAGENT calls a new utility CASCKREX to check that REXX is initialized on the client workstation. It returns 0 if REXX is initialized and otherwise 1.

—— **MPTS 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.

4.1.3.6 CASDELET

Deletes the STARTUP.CMD and removes the CONFIG.SYS statements enforced by CASINSTL execution.

CASDELET Syntax

```
CASDELET /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 CASDELET's log file.

Usually IFSDDEL and CASDELET are called in the last execution of the LCU command file after all products are installed to the client.

CASDELET Example

```
CASDELET /TU:C  
          /PL:X:\DLL\CONNECT;X:\IMG\LCU  
          /L1:L:\lcu\log1.log
```

```

x.casdelet = 23                                /* structure index */
x.23.name='LCU Cleanup'                        /* product name */
x.23.statevar = ''                             /* state variable name */
x.23.instprog = 'x:\img\lcu\casdelet           ', /* install program */
                '/P1:x:\d11\CONNECT;x:\img\lcu; ', /* - delete from libpath*/
                '/tu:' || bootdrive             /* - config.sys locat. */
x.23.rspdir = ''                               /* no auto selection */
x.23.default = ''

where x: is the shared drive to the code server's subdirectory \CID
      Access to x: is usually defined as 'Read only'
and y: is the shared drive to the code server's subdirectory \CID\LOG
      Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 19. Extract of an LCU Client Command File Illustrating CASDELET Program Invocation

4.1.4 Communications Manager/2

CM/2 V1.11 uses the same installation program, CMSETUP, for both attended interactive configuration and installation as well as redirected response file driven installation. CMSETUP is explained in detail in the online CM/2 Command Reference.

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 CM/2 V1.11 installation.

If you already have a working OS/2 system without some sort of redirector you need to boot on diskettes. First the client is booted on diskettes to get a redirected drive to the code server and redirector code is installed on the client. The user at the client is asked to remove the diskette and the client is automatically rebooted. Then it continues with CM/2 V1.11 installation and is rebooted again. If a temporary redirector was installed it can be removed as the last step.

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 Installation

```
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).

For more information on response files and remote installation, refer to *IBM Communications Manager/2 Version 1.1 Network Administration and Subsystem Management Guide* and to 3.2.2, “Communications Manager/2” on page 52.

CMSETUP allows you to request the following installation actions based on the specified response file:

- Install IBM Communications Manager/2 Version 1.11 on a drive with no Communications Manager.
- Upgrade a previous Communications Manager release to IBM Communications Manager/2 Version 1.11 (including installation and configuration).
- Configuration change with installation of additional features if necessary.
- Configuration change without installation.
- Re-install Communications Manager.
- Remove communication features (based on the default configuration).

/S Fully qualified path to CM/2 V1.11 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 CM/2 V1.11**
- No check will be made to determine if it contains Extended Services 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 an Extended Services 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 CM/2 V1.11 diskette images are located on the code server. So CMSETUP does not need /S to be explicitly set.

CMSETUP Example

```
X:\img\cm2111\CMSETUP /CR
                        /L1 L:\cm2111\cmrinst.log
                        /L2 L:\cm2111\cmaudit.log
                        /R X:\rsp\cm2111\clientx.rsp
```

CMSETUP is invoked from the redirected drive X:imgcm2111. The response file, 'clientx.rsp' is a response file made for CM/2 V1.11. The log files will be logged on the redirected drive L:cm2111.

```
x.cm111 = 16                                /* structure index */
x.16.name='CM/2 1.11'                        /* product name */
x.16.statevar = 'CAS_' || x.16.name          /* state variable name */
x.16.instprog = 'x:\img\cm2111\cmsetup', /* install program */
                        /cr, /* - current flag */
                        /, /* if migration use */
                        /mg <path> <filename> /*
                        /, /* /KL keylock password */
                        /11:l:\cm2111\ || client || '.11', /* install log */
                        /12:l:\cm2111\ || client || '.12', /* audit trail log*/
                        /r, /* - response file */
x.16.rspdir = 'x:\rsp\cm2111'                /* response file dir. */
x.16.default = 'mod3270.rsp'                 /* default rsp file */
```

where x: is the shared drive to the code server's subdirectory \CID
Access to x: is usually defined as 'Read only'
and y: is the shared drive to the code server's subdirectory \CID\LOG
Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

Figure 20. Extract of an LCU Client Command File Illustrating CMSETUP Program Invocation

Be sure to reboot the client after CMSETUP is executed so that locked files are processed.

4.1.4.1 Installation of CM/2 V1.11 Distributed Feature

The CM/2 V1.11 Distributed Feature places most of the CM/2 code onto a file server. It has been tested using IBM LAN Server V2.0 or later and Novell NetWare Version 3.11 or later. Installation of the CM/2 server depends on how it will be used:

- A dedicated CM/2 server, as it would be when using Novell NetWare Server, will be installed using the **CMIMAGE** utility combined with the **/U** option. **CMIMAGE /U** unpacks the zipped code into the directory pointed to by the CMIMAGE utility.

- A CM/2 server that will run CM/2 for its own use, as it may be used on an IBM LAN server. The server will be installed using the CMSETUP utility with a response file including the CMServer keyword.

The CM/2 Distributed Feature client workstation is installed using the **CMLAN** utility combined with a response file having the **CMWorkStationType** keyword value set to **4**.

CMLAN Example

```
X:\img\cm2111\CMLAN  /L1 L:\cm2111\cmrinst.log
                      /L2 L:\cm2111\cmaudit.log
                      /R X:\rsp\cm2110\clientx.rsp
```

4.1.5 Personal Communications/3270 for OS/2

Personal Communications/3270 for OS/2 is the successor program for the 3270/5250 Emulators of CM/2 V1.11. It is only used for the Emulation functions and fundamental APPC Communications Support. All Gateway functions are implemented in the CM Server V4.0 Here we will briefly explain the CID Installation of the product. The technical description is available in the Online documentation of PC/3270 for OS/2 V4.1. 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 Personal Communications/3270 for OS/2 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.

INSTALL Syntax For Response File Driven Installation

```
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>
```

You must enter either a colon or a space after the parameters. You do not need to enter file extensions.

/A Centralized installation

Use this parameter to install PC/3270 for OS/2 V4.1 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 errorlog

Specifies the complete drive, path and filename for the errorlog file for this installation.

- /L2** **Complete filename of the historylog**
- Specifies the complete drive, path and filename for the historylog 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 standalone PC/3270 for OS/2 V4.1. If /M:C, the migration is to PC/3270 for OS/2 V4.1 using CM/2 communication.
- /N** **Centralized installation**
- Use this parameter when installing PC/3270 for OS/2 V4.1 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!

— PC/3270 for OS/2 V4.1 Example —

```
X:\img\PCOS2V41\INSTALL
      /S X:\img\PCOS2V41
      /T C:\PCOMOS2
      /G X:\img\PCOS2V41\RSP
      /L1 L:\PCOS2V41\cmrinst.log
      /L2 L:\PCOS2V41\cmaudit.log
      /R X:\rsp\PCOS2V41\clientx.rsp
```

```

x.PCOMOS2V41 = 10
x.10.name='PC/3270 for OS/2 Version 4.1'
x.10.statevar = 'CAS_' || x.10.name
x.10.instprog = 'x:\img\pcos2v31\install',
               's:x:\img\pcos2v41',
               'T:C:\pcomos2',
               'l1:l:\pcos2v41\' || client || '.err ',
               'l2:l:\pcos2v41\' || client || '.his ',
               'g:x:\img\pcos2v41\rsp',
               'r:'
x.10.rspdir = 'x:\rsp\pcos2v41'
x.10.default = 'PCOMOS2.RSP'

/* structure index */
/* product name */
/* state variable name */
/* install program name */
/* source directory */
/* - Target directory */
/* errorlog file */
/* history logfile */
/* include file directory */
/* - response file flag */
/* response file directory */
/*default response file name */

where x: is the shared drive to the code server's subdirectory \CID
        Access to x: is usually defined as 'Read only'
and L: is the shared drive to the code server's subdirectory \CID\LOG
        Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 21. Extract of an LCU Client Command File Illustrating INSTALL Program Invocation

A complete description of the response file keywords comes with the product. It is stored in the file PCSREF.INF under the subdirectory INFO. You can find a copy of this file on the Sample CDROM in the directory MISC.

4.1.5.1 Installation of PC/3270 for OS/2 V4.1 on a Server

The installation PC/3270 for OS/2 V4.1 as a Server application has changed compared to the installation of CM/2 V1.11. The installation is done with the same INSTALL Command. You have to use the /A Option to do the installation on the redirected Server drive. The kind of emulation has to be chosen during Installation time. The files are transferred to the server drive and can be accessed from the clients. In the documentation this Server type is called a Server Type II. This installation can be response file driven too.

PC/3270 for OS/2 V4.1 Example for Server Installation

```

X:\img\PCOS2V41\INSTALL /CR
                        /A
                        /L1 L:\PCOS2V41\cmrinst.log
                        /L2 L:\PCOS2V41\cmaudit.log
                        /R X:\rsp\PCOS2V41\clientx.rsp

```

The clients connected to this server can only use the emulation that was chosen for the server.

To get access to this server PC/3270 for OS/2 V4.1 has to be installed on the clients with the /N Option. The installation procedure is already described.

4.1.6 Communications Server for OS/2 Warp Version 4.0

The new CM Server V4.0 includes all gateway functions and is used for SNA communications. The emulator for 3270/5250 emulations are no longer part of the CM Server V4.0. They are separately delivered in the PC/3270 for OS/2 V4.1. There are many different gateway functions included in this product but here we will only describe the installation of CM Server V4.0 in a CID environment. Compared to CM/2 V1.11 the installation of the product has not changed. The installation program is still called CMSETUP and the parameters for the automated installation are still the same. The installation creates the CM Server V4.0 folder on the desktop with the difference that in this folder there are two additional folders included for administrator and problem determination tasks. So there are not so many objects in the folder and the different functions can be found easily. The installation of CM Server V4.0 is a PM program so be sure to have a fully functional OS/2 Warp Connect running for the installation. All the other prerequisites and parameter descriptions are mentioned in 4.1.4, "Communications Manager/2" on page 108.

Complete documentation comes with the CM Server V4.0 CDROM and includes many INF-files in the subdirectory BOOKSINF.

```

x.CMSRV4 = 11                                /* structure index */
x.11.name='CM Server for OS/2                 /* product name */
x.11.statevar = 'CAS_' || x.11.name           /* state variable name */
x.11.instprog = 'x:\img\cmsrv4\cmsetup        /* install program name */
              's:x:\img\cmsrv4              /* - source directory */
              'cr                          /* - current response flag */
              'l1:l:\cmsrv4\ ||client || '.11 ' /* error log file */
              'l2:l:\cmsrv4\ ||client || '.12 ' /* history log file*/
              'g:x:\img\cmsrv4\rspfile        /* include file directory */
              'r:'                          /* - response file flag */
x.11.rspdir = 'x:\rsp\cmsrv4'                 /* response file directory */
x.11.default = 'CMSRVGW.RSP'                 /* response file name */

```

where x: is the shared drive to the code server's subdirectory \CID
Access to x: is usually defined as 'Read only'
and L: is the shared drive to the code server's subdirectory \CID\LOG
Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

Figure 22. Extract of an LCU Client Command File for CM Server Install

4.1.7 Database 2 for OS/2

All varieties of IBM DATABASE 2 for OS/2 Version 2.11 use identical utility programs (from Software Installer) for installation and preparation of response files:

- INSTALL.EXE
- DB2RESP.EXE

In our lab we performed tests with

- DB2/2 V2.11 Single-User Version
- DB2/2 V2.11 Server Version
- DB2 Client Application Enabler/2 Version 2.11

INSTALL Syntax

```
INSTALL /R:<response file>
/X
/A:<action>
/G:<general include path>
/P:<product name>
/S:<source path>
/T:<install target directory>
/L1:<error log file>
/L2:<history log file>
```

Option /X is required for unattended installation as well as the /R and /A parameters.

/R: Fully qualified path and name of response file

/X Unattended execution (mandatory for CID)

/A: Action to be performed (mandatory)

- **D** Delete
- **I** Install (Default)
- **R** Restore
- **U** Update

/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.

/P: Product name

Specifies the name of the product (Server, Single-User, Software Developer's Kit, ...) to be installed. Make sure that the spelling of the product name is correct. Be aware that the product names are language dependent.

Note: You must specify this option only if you are installing from a CD-ROM. Further information about this (for example the required exact spelling of the product names) can be found in the *Database 2 for OS/2 Version 2.1.1 Installation and Operation Guide*, S20H-4785.

/S: Fully qualified path to the DB2/2 images

If omitted, the installation program assumes that the files reside in the same directory from which it is running.

Remember to point to the correct directory. Assuming that the proposed CID directory structure is used it would be:

- <drive>:CIDIMGDB2V211DB2SRVR
for DB2/2 V2.11 Server Version
- <drive>:CIDIMGDB2V211DB2SU
for DB2/2 V2.11 Single-User Version
- <drive>:CIDIMGDB2V211DB2CAE2
for DB2 Client Application Enabler/2 Version 2.11

/T: Installation target directory

/L1: Error log name

/L2: History log name

Known problems for INSTALL: Remote logging fails

There is a known problem installing DB2/2 via CID, if the log files are written to a remote drive. Further information can be found in APAR JR08659 as well as in various DB2/2 or CID related fora. Here is a description of the symptoms:

We experienced this problem using DB2/2 V2.11 and NetView DM/2 V2.1, but other versions or CID products are likely to fail as well.

The symptoms are (extract from NetView DM/2 V2.1 MESSAGE.DAT):

```
=====
** NetView DM/2 logged at 08:04:06 (AM) 02-20-1996 **
ANX1315: (I) Invoking External Program: 'X:\IMG\DB2SU\INSTALL.EXE' With Parms:
'/X /A:I /S:X:\IMG\DB2SU /R:X:\RSP\DB2SU\TEST.RSP
/L1:W:\LOG\DB2SU\TEST.L1
/L2:W:\LOG\DB2SU\TEST.L2'
=====
** NetView DM/2 logged at 08:08:05 (AM) 02-20-1996 **
ANX0253: (E) The External Program 'X:\IMG\DB2SU\INSTALL.EXE' failed with exit
code '000e'. Refer to the log file(s) produced by the external program for
additional details on the error. Check the Change File Profile to locate them.
=====
** NetView DM/2 logged at 08:08:19 (AM) 02-20-1996 **
ANX1311: (E) This Exit Code is not an architected code for the CID products.
=====
** NetView DM/2 logged at 08:08:24 (AM) 02-20-1996 **
```

ANX0210: (W) Network access is denied.

=====

** NetView DM/2 logged at 08:08:27 (AM) 02-20-1996 **

ANX0264: (E) Connection request to CC Server rejected.'

=====

** NetView DM/2 logged at 08:08:28 (AM) 02-20-1996 **

ANX0247: (I) The CDM Agent on the CC Client has ended because unable to link the CC Server.

=====

The problem is not caused by DB2/2, but by Software Installer installation utilities. The problem should be fixed with Software Installer Version 1.4. Until this is available, you may bypass the problem by writing the log files to a local drive of the target machine.

Installation Example for DB2 Client/Server

```
X:\IMG\DB2V211\DB2SRVR\INSTALL
      /R:X:\RSP\DB2V211\DB2SRVR\$(WorkStatName).RSP
      /X
      /A:I
      /G:X:\RSP\DB2V211\DB2SRVR
      /S:X:\IMG\DB2V211\DB2SRVR
      /T:C:\SQLLIB
      /L1:C:\DB2SRVR.L1
      /L2:C:\DB2SRVR.L2
```

To install a database server INSTALL is invoked from the X:IMGDB2V211DB2SRVR directory, where the DB2/2 V2.11 Server Version diskette images are stored. The response file is read from the X:RSPDB2V211DB2SRVR directory and the log files are written to the local C: drive of the installation target machine. This is due to the INSTALL program's remote logging problems mentioned above.

```

x.db2su = 12                                /* structure index */
x.12.name='DB2/2 Server V2.1.1'             /* product name */
x.12.statevar = 'CAS_' || x.12.name          /* state variable name */
x.12.instprog = 'x:\img\db2srvr\install      /* install program */
                '/11:1:\db2srvr\' || client || '.log ', /* - log file */
                '/r:'                          /* - response file flag */
x.12.rspdir  = 'x:\rsp\db2srvr'              /* response file dir. */
x.12.default = 'moddb2su.

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
and y: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'

```

Figure 23. Extract of an LCU Client Command File Illustrating DBCID Program Invocation

The installation of other DB2/2 products works exactly the same way as the server installation described above. The differences are:

- Different response files
- Different product directories

4.1.8 LAN Requester/Server

LAN Server V4.0 and LAN Server V5.0 uses LANINSTR for installation from a redirected drive. The installation can be:

- Attended remote installation
- Lightly attended remote installation
- Unattended remote installation

The parameters specified and the environment will make the difference as to which mode of installation will be performed. The only one we will discuss here is the unattended installation from LANINSTR's point of view.

To invoke **LANINSTR** OS/2 V2.0 or later must be running on the target workstation. The OS/2 must be running PM and the Workplace Shell. So after an OS/2 installation the client must be rebooted before the installation of LAN Server/Requester is done. The OS/2 maintenance system (non-PM) will not suffice.

LANINSTR Syntax - Lightly Attended and Unattended Mode

```
X:\img\LS50\LANINSTR /REQ | /SRV
                        /R:<response file>
                        /G:<general path>
                        /L1:<log file 1>
                        /L2:<log file 2>
```

The example above will run LANINSTR from the LAN Server V4.0 Advanced. If another version should be installed be careful to invoke LANINSTR from the correct source code directory.

/REQ **Requester installation**

/SRV **Server installation**

Either /REQ or /SRV has to be set. If /SRV is chosen the requester code is installed automatically.

/R: **Fully qualified path and name of response file**

/G: **Fully qualified path of an INCLUDE file directory**

This is the drive and path used to locate an included group response file.

/L1: **Fully qualified path and file name of error log.**

/L2: **Fully qualified file name of history log**

When the installation is completed, check the error log and the history log at the code server. These files will also be written to the client workstation.

The error log file at the client workstation is named

d:OS2INSTALLIBMLANER.LOG. The history log file is named

d:OS2INSTALLIBMLSHST.LOG. In this case *d* is the drive letter where the base OS/2 system is installed.

```

x.lanreqa = 10                                /* structure index */
x.10.name='LAN Requester 5.0 Advanced'         /* product name */
x.10.statevar = 'CAS_' || x.10.name            /* state variable name */
x.10.instprog = 'x:\img\ls501\laninstr'        /* install program */
                                                /* - install a requester */
                                                /* error log file */
          ' /11:l:\ls50\' || client || '.L1    ',
          ' /12:l:\ls50\' || client || '.L2    ', /* history log file */
          ' /r:x:\rsp\ls50\' || client || '.rsp' /* response file */
x.10.rspdir = ''                               /* no response file dir. */
x.10.default = ''                             /* no default rsp file */

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
       and l: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 24. Extract of an LCU Client Command File Illustrating LANINSTR Program Invocation

LANINSTR gives a return code indicating that a reboot of the client should be performed after LANINSTR is finished. It is up to the software distribution manager program to check for this.

4.1.9 TCP/IP V3.0

INSTALL.EXE is used to install TCP/IP V3.0. The install program for TCP/IP can be invoked with command line parameters as shown in the syntax overview below. It is also supported to use response file keywords to specify the installation and configuration. The TCP/IP V3.0 is shipped with OS/2 Warp Connect. It include the following packages:

- Base TCP/IP Applications
- Feature TCP/IP Applications: WE/2, NR/2, Gopher and Internet Dialers
- DOS/Windows Access
- UltiMail Lite

To do a CID installation of TCP/IP V3.0 you have to install and setup MPTS first. It must be the MPTS shipped with OS/2 Warp Connect. The values configured in the PROTOCOL.INI are read by the TCP/IP V3.0 installation program and the configuration is updated according to the values defined in the PROTOCOL.INI.

<i>Table 7 (Page 1 of 3). TCP/IP V3.0 INSTALL Parameters</i>		
Parameter	Description	Corresponding Responsefile Keyword
/b	Turns off the beep that accompanies the prompt for the next diskette.	none
/r:filename	Fully qualified filename of the TCP/IP response file.	RSP_PATH
/ip:ip_address	Specifies the internet address of the workstation. The format is mmm.nnn.nnn.nnn.	IPADDR
/nm:netmask	Specifies the subnet mask of the workstation. The format is nnn.nnn.nnn.nnn	
/rt:router_address	Specifies the internet address of the default router. The format is nnn.nnn.nnn.nnn	ROUTE
/h:hostname	Specifies the host name of the workstation	HOSTNAME
/s:source_path	Specifies the complete path on the code server that contains the diskette images for TCP/IP V3.0 The default is the path from which the INSTALL command was issued.	SOURCE_PATH
/t:target_path	Specifies the complete path on the workstation to which TCP/IP V3.0 is to be installed.	TARGET_PATH
/lp:laps_path	Specifies the complete path on the code server that contains the MPTS.EXE	LAPS_EXE_PATH
/lr:laps_response_file	Specifies the complete path on the code server that contains the MPTS response file.	LAPS_RSP_FILE

<i>Table 7 (Page 2 of 3). TCP/IP V3.0 INSTALL Parameters</i>		
Parameter	Description	Corresponding Responsefile Keyword
/tu:boot_drive	Specifies the drive from which your workstation starts (bootdrive)	
/sf-	Specifies to add TCPSTART to the STARTUP.CMD file instead of adding it to the startup folder. If you omit this parameter, TCPSTART is added to the startup folder.	STARTUP_FOLDER
/srv: "service1,...,servicen"	Specifies one or more TCP/IP services to be included in the TCPSTART.CMD for automatic startup when TCP/IP initializes. The services are comma separated.	TCP_SERVICES
/l1:pathfilename.extension	Specifies the fully qualified filename for the TCP/IP installation error log.	LOG_PATH (the path where the log-files are stored)
/l2:pathfilename.extension	Specifies the fully qualified filename for the TCP/IP installation history log.	LOG_PATH (the path where the log-files are stored)
/c	Causes INSTALL to make the necessary changes to the CONFIG.SYS, without installing TCP/IP V3.0. This is usefull if your CONFIG.SYS is erased during the installation of OS/2	CONFIG_SYS

<i>Table 7 (Page 3 of 3). TCP/IP V3.0 INSTALL Parameters</i>		
Parameter	Description	Corresponding Responsefile Keyword
/a-	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.	ATTENDED

The LOG_PATH typically points to a path on the code server so that a network administrator can access the log files if a failure occurs. A default TCP/IP installation response file comes with OS/2 Warp Connect and is called DEFAULT.RSP. It is on the Sample CDROM also. You can add the above parameters at the end of the file to prepare it for your environment.

We only used the parameters for an automated installation in a CID environment. As it is optional to use either the invocation parameters or the response file keywords, we recommend using the response file keywords to specify the client-specific details of installation and configuration. It is easier to maintain only one product definition and client-specific response files than to create different product definitions, probably including client-specific response files, for every workstation.

— INSTALL Example for Usage with CID Installs —

```
X:\img\tcpip\install
    /S:X:\img\tcpip30
    /T:C:\TCPIP30
    /TU:C:/
    /L1:L:\tcpip30\client1.11
    /L2:L:\tcpip30\client1.12
    /R:X:\rsp\tcpip30\client1.rsp
```



```

x.tcpip30 = 17                                /* structure index */
x.17.name='TCP/IP V3.0'                      /* product name */
x.17.statevar = 'CAS_' || x.17.name          /* state variable name */
x.17.instprog = 'x:\img\tcpip30\install', /* install program */
               ' /s:x:\img\tcpip30',        /* - source directory */
               ' /t:' || bootdrive || '\tcpip30', /* - target directory */
               ' /tu:' || bootdrive || '\',    /* - config.sys location */
               ' /r:x:\rsp\tcpip30\' || client || '.rsp' /* - response file*/
x.17.rspdir = ''                             /* no auto selection */
x.17.default = ''

where x: is the shared drive to the code server's subdirectory \CID
       Access to x: is usually defined as 'Read only'
and l: is the shared drive to the code server's subdirectory \CID\LOG
       Access to l: is usually defined as 'Read/Write'
(see Figure 9 on page 41)

```

Figure 25. Extract of an LCU Client Command File Illustrating INSTALL Program Invocation

4.1.10 Product Installation Order

There is no definite order that absolutely has to be followed, but we recommend that you bear the following sequence in mind. Apply the latest Service Pak to a product as soon as possible after the product is installed.

OS/2	Kind of obvious that this should come first.
LAN transport	So the client can connect to the code server again to continue installation.
OS/2 Service Pak	If there is one you have to use the FSERVICE program to apply the Service Pak OS/2 system.
LAN Server/Requester	If the client will use a LAN.
Communications program	Other applications may have the communication as a prerequisite.
Database	Which may have communication as a prerequisite and itself be prerequisite for other applications.
Other applications	CID-enabled or those that can be "cloned" if they are installed on the code server.

The sample CONNECT.CMD on the sample code CDROM installs the products in the following order:

1. OS/2 Warp Connect
2. MPTS

3. "Thin" LAN requester to be able to connect back
4. LAN Requester V5.0 or TCP/IP V3.0
5. CM/2 V1.11 or PC/3270 for OS/2 V4.1 or CM Server V4.0
6. DB2/2

4.2 Installation Commands for Products that Are Not CID-Enabled

4.2.1 Installation of Novell NetWare Workstation for OS/2 V2.01

Note

The following chapter is only valid for the Netware Requester shipped with Netware Version 3.12. A new Netware Client is shipped together with OS/2 Warp Connect but it is not CID enabled. A CID Environment as described in this chapter is not running with Netware Version 4.10. For this Netware Release the NetView DM for Netware should be used to set up a CID environment.

The Novell NetWare Workstation for OS/2 V2.01 is referenced as NetWare requester in this chapter. As the NetWare requester is not yet CID-enabled, additional procedures are needed to get the requester installed on a client machine. These procedures will be described in detail in this section and they can be found in the NetWare subdirectory of the sample code CDROM. They have to be copied to the IMGSAAMPLENetWare subdirectory to be available for the sample installation provided with this book.

For the installation of the NetWare requester the product files have to be copied to the code server. Please refer to 16.1.10, "Loading NetWare Requester Files" on page 395 for information on that task. As the files of an installed version of NetWare requester are used, it is imperative that the level of the code for the NetWare requester of this installed version is the one that will be installed on the clients.

The **NWINST.CMD** procedure is used to copy the NetWare requester files to the client and to do the necessary changes to the CONFIG.SYS, that is adding the new directory NetWare to the PATH, DPATH and LIBPATH statements and adding the NetWare device statements. In our sample installation, the network driver ODI2NDI.OS2 supplied by MPTS is used for the NetWare requester. It is installed during the MPTS installation. There is an MPTS response file for the NetWare requester install supplied on the

sample code CDROM. Remember to use LAPSRSP as described in 3.2.4, “MPTS Response File” on page 58 if you want to create your own response file for LAPS. LAPSRSP.EXE has to be executed on a workstation running the same driver configuration you want to install.

Additionally, the NWINST procedure will set up the required environment on the client workstation to reattach to the code server, after OS/2 Warp Connect (or OS/2 V2.11) has been installed, and a reboot done.

As the NWINST procedure just copies the NetWare requester files to the client workstation, there will be no icon created on the client’s desktop. There is a procedure called NWICON.CMD that creates the Novell folder on the client’s desktop. This procedure cannot be invoked during the initial install because it uses REXX functions that need PM to be available. This procedure can be invoked either as a user exit from one of the installs that follow the initial OS/2 install or a separate product definition and install command can be used. The product definition can be found in the default LCU command file provided on the sample code CDROM. The install part shown in 4.4.7.4, “NetWare LCU Command File” on page 168 integrates the NWICON.CMD after the DB2/2 install. NWICON.CMD needs two parameters; target drive and directory name. The code of NWICON.CMD can be found in the NetWare subdirectory of the sample code CDROM and it has to be copied to the IMGSAAMPLENetWare subdirectory to be executed during NWINST.

The following figure shows the NWINST procedure.

```

/* ----- */
/* NWINST.CMD */
/* */
/* REXX procedure which will perform the following steps: */
/* */
/* 1. Copy the NetWare files from the server to the clients \NetWare */
/* subdirectory. */
/* 2. Update existing lines (PATH, DPATH and LIBPATH) in the client */
/* CONFIG.SYS. */
/* 3. Add new lines to the client CONFIG.SYS for the required device drivers. */
/* */
/* This procedure is invoked from the LCU command file. */
/* It assumes that the LTS diskette will be left in the drive until the end */
/* of this procedure in order to copy the file ENV_VARS.CMD. */
/* ----- */
address cmd
'echo off'
env='OS2ENVIRONMENT' /* Access the OS2 environment */
Cltdrv='C:' /* Default OS/2 Drive */
NWDrv='C:' /* Sets drive letter for NetWare */
/* The NWDrv letter is the drive where NetWare will be */
/* installed. This will typically be the same as for OS2 */
/* but you may specify any valid drive letter. */

do while lines(Cltdrv||'\config.sys') /* Do until end of file */
it = linein(Cltdrv||'\config.sys') /* Read first line */
it = translate(it) /* Make everything UPPERCASE */

if substr(it,length(it),1) == ';' /* check for semicolon at lineend */
then sc = ''
else sc = ';'

if pos( 'SET PATH' , it ) <> 0 then do /* If SET PATH is in line */
it = it||sc||NWDrv||'\NetWare;' /* Add (NWDrv);\NETWARE */
end

if pos( 'SET DPATH' , it ) <> 0 then do /* If SET DPATH is in line */
it = it||sc||NWDrv||'\NetWare;X:\IMG\LCU;' /* Add (NWDrv);\NETWARE */
/* and DPATH to IMAGES */
/* for CID install */

if pos( 'LIBPATH' , it ) <> 0 then do /* If LIBPATH is in line */
it = it||sc||NWDrv||'\NetWare;X:\DLL\OS2V211;' /* Add (NWDrv);\NETWARE */
/* and X:\DLL\OS2V211 for */
/* CID install of OS/2 */
/* Version 2.11 */

call lineout Cltdrv||'\config.new', it /* Write line to config.new */
end

```

Figure 26 (Part 1 of 2). NWINST.CMD Procedure

```

/* - - - - - */
/* close the files */
/* - - - - - */
call stream CltDrv||'\config.new','c','close'
call stream CltDrv||'\config.sys','c','close'
'copy' CltDrv||'\config.new' CltDrv||'\config.sys'
'del' CltDrv||'\config.new'

/* - - - - - */
/* The following lines add the required device statements to the CONFIG.SYS */
/* for the NetWare network. Includes TOKEN RING drivers only! */
/*
/* The NetWare Driver ODI2NDI.SYS will be installed via the LAPS Install. */
/* See the LAPSNW.RSP response file on the sample code disk for further */
/* information. */
/* - - - - - */

call lineout CltDrv||'\config.sys',' '
call lineout CltDrv||'\config.sys','REM'
call lineout CltDrv||'\config.sys','REM Beginning of NetWare device statements'
call lineout CltDrv||'\config.sys','REM'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\LSL.SYS'
call lineout CltDrv||'\config.sys','RUN=' NWDrv||'\NETWARE\DDAEMON.EXE'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\ROUTE.SYS'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\IPX.SYS'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\NWREQ.SYS'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\SPX.SYS'
call lineout CltDrv||'\config.sys','RUN=' NWDrv||'\NETWARE\SPDAEMON.EXE'
call lineout CltDrv||'\config.sys','IFS=' NWDrv||'\NETWARE\NWIFS.IFS'
call lineout CltDrv||'\config.sys','RUN=' NWDrv||'\NETWARE\NWDAEMON.EXE'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\VIPX.SYS'
call lineout CltDrv||'\config.sys','DEVICE=' NWDrv||'\NETWARE\VSHELL.SYS'
call lineout CltDrv||'\config.sys','REM'
call lineout CltDrv||'\config.sys','REM - NetWare Requester statements END --'
call lineout CltDrv||'\config.sys'

/* - - - - - */
/* This section will make the NWDrv NetWare directory and copy the NetWare */
/* requester files into it from the server */
/* - - - - - */

'md' NWDrv||'\NetWare'
'copy' X:\IMG\NetWare\*. * 'NWDrv||'\NetWare'

/* - - - - - */
/* Now the required additional files to re-connect to the CID server are */
/* copied */
/* - - - - - */

'copy' X:\IMG\SAMPLE\NetWare\startnw.cmd 'NWDrv||'\startup.cmd'
'copy' a:\env_vars.cmd c:\
'copy' a:\startrfi.cmd c:\

```

Figure 26 (Part 2 of 2). NWINST.CMD Procedure

The **NWDELETE.CMD** procedure is used to remove the procedures and files used to reattach to the code server, and clean up the root drive on the

client's workstation. It works similar to the IFSDEL and CASDELET procedures during an installation using SRVIFS. A NWSEED directory is created and all procedures and files to make the required connection back to the code server are copied to it. This will allow the client to once again gain access to the code server in order to obtain any software maintenance, without having to use the LAN transport system diskettes.

The following figure shows the NWDELETE procedure.

```

/* ----- */
/* NWDELETE.CMD */
/*
/* This Procedure deletes the files used to connect to the Novell Server
/* during the CID Installation and saves several files for the NWSEED
/* Procedure.
/* ----- */

Cltdrv='C:' /* Default OS/2 Drive */

'md c:\nwseed' /* Create Novell seed directory */

'copy c:\startup.cmd c:\nwseed\startup.nw' /* Remove Startup Novell command */
'copy x:\img\sample\crenvvar.exe c:\nwseed' /* file from the root directory */
'copy x:\img\sample\netware\nwseed.cmd c:\'
'del c:\env_vars.cmd'

/* ----- */
/* Delete all the CAS statements from the CONFIG.SYS file
/* ----- */

do while lines(Cltdrv||'\config.sys') /* Do until end of file */
  it = linein(Cltdrv||'\config.sys') /* Read first line */
  it = translate(it) /* Make everything UPPERCASE */
  if pos('SET CAS',it) <> 0 then do
    it=''
  end
  if it <> '' then do
    call lineout Cltdrv||'\config.new', it /* Write line to config.new */
  end
end

/* ----- */
/* close the files */
/* ----- */

call stream Cltdrv||'\config.new','c','close'
call stream Cltdrv||'\config.sys','c','close'
'copy' Cltdrv||'\config.new' Cltdrv||'\config.sys'
'del 'Cltdrv||'\config.new'

/* ----- */
/* Remove the call to Novell NetWare startup file from the STARTUP.CMD
/* ----- */

do while lines(Cltdrv||'\startup.cmd') /* Do until end of file */
  it = linein(Cltdrv||'\startup.cmd') /* Read first line */
  it = translate(it) /* Make everything UPPERCASE */

  if pos('STARTRFI.CMD',it) <> 0 then do
    it=''
  end
  if it <> '' then do
    call lineout Cltdrv||'\startup.new', it /* Write line to config.new */
  end
end

```

Figure 27 (Part 1 of 2). NWDELETE.CMD Procedure

```

/* - - - - - */
/* close the files */
/* - - - - - */

call stream CltDrv||'\startup.new','c','close'
call stream CltDrv||'\startup.cmd','c','close'
'copy' CltDrv||'\startup.new' CltDrv||'\startup.cmd'
'del' 'CltDrv||'\startup.new'

/* - - - - - */
/* Save the CONFIG.SYS for the NWSEED Procedure */
/* - - - - - */

'copy c:\config.sys c:\nwseed\config.nw'

```

Figure 27 (Part 2 of 2). NWDELETE.CMD Procedure

The **NWSEED.CMD** procedure is used to reattach the client workstation to the code server using the procedures and files found in the NWSEED subdirectory. For a detailed description of seed and maintenance scenarios please refer to Chapter 5, "Maintenance and Service" on page 183.

```

/* - - - - - */
/* NWSEED.CMD */
/* */
/* Create attach to Novell Server for SEMAINT using the copies saved by the */
/* NWDELETE.CMD */
/* - - - - - */
'copy c:\config.sys c:\nwseed\config.os2'
'copy c:\nwseed\config.nw c:\config.sys'
'copy c:\startup.cmd c:\nwseed\startup.os2'
'copy c:\autoexec.bat c:\nwseed\autoexec.os2'
'copy c:\nwseed\startup.nw c:\startup.cmd'
'copy c:\nwseed\crenvvar.exe c:\'
'c:\os2\setboot /ibd:c' /* Reboot invocation */

```

Figure 28. NWSEED.CMD Procedure

The **STARTNW.CMD** file contains the call to the STARTRFI.CMD procedure. Please refer to 12.4.2, "Adding LAN Transport System to Client diskettes" on page 322 for detailed information on STARTRFI.CMD. It is copied to the root directory of the client workstation during the NWINST.CMD procedure and renamed to STARTUP.CMD.

```

Call STARTRFI.CMD
EXIT

```

Figure 29. STARTNW.CMD File

This **NWPREP.CMD** procedure will edit the temporary CONFIG.SYS created by SEMAINT in order to add the driver statements that allow the client to reattach to the code server after the SEMAINT procedure has completed.

```

/* - - - - - */
/* NWPREP.CMD */
/*
/* This REXX procedure which will change the CONFIG.SYS that was created by
/* SEMAINT in order to connect back to the code server after booting from the
/* C:\SERVICE subdirectory.
/*
/* This procedure is invoked from the LCU command file NWCLIENT.CMD.
/* - - - - - */
address cmd
'echo off'
env='OS2ENVIRONMENT' /* Access the OS2 environment */
Cltdrv='C:' /* Default OS/2 Drive */
NWDrv='C:' /* Sets drive letter for NetWare and LAPS */
ComDrv='C:' /* The NWDrv letter is the drive where NetWare was
/* installed. The ComDrv letter is the drive where LAPS
/* was installed. These letters may be changed. */

do while lines(Cltdrv||'\config.sys') /* Do until end of file */
it = linein(Cltdrv||'\config.sys') /* Read first line */
it = translate(it) /* Make everything UPPERCASE */

if substr(it,length(it),1) == ';' /* check for semicolon at lineend */
then sc = ''
else sc = ';'

if pos('SET OS2_SHELL',it) <> 0 then do /* If SET-OS2_Shell is in line */
it = it '/' K C:\STARTRFI.CMD' /* add call for startrfi.cmd */
end

if pos('SET PATH', it) <> 0 then do /* If SET PATH is in line */
it = it||sc||NWDrv||'\NetWare;' /* Add (NWDrv);\NETWARE */
end

if pos('SET DPATH', it) <> 0 then do /* If SET DPATH is in line */
it = it||sc||NWDrv||'\NetWare;X:\IMG\LCU;C:\IBMCOM;' /* Add (NWDrv);\NETWARE and DPATH */
end /* to IMAGES for CID install */

if pos('LIBPATH', it) <> 0 then do /* If LIBPATH is in line */
it = it||sc||NWDrv||'\NetWare;X:\DLL\OS2V20;C:\IBMCOM\DLL;' /* Add (NWDrv);\NetWare and */
end /* X:\DLL for CID install */

call lineout Cltdrv||'\config.new', it /* Write line to config.new */
end

/* - - - - - */
/* close the files */
/* - - - - - */
call stream Cltdrv||'\config.new','c','close'
call stream Cltdrv||'\config.sys','c','close'
'copy' Cltdrv||'\config.new' Cltdrv||'\config.sys'
'del' Cltdrv||'\config.new'

```

Figure 30 (Part 1 of 2). NWPREP.CMD Procedure

```

/* ----- */
/* The following lines add the required device statements to the CONFIG.SYS */
/* for the NetWare network, as it was done during the installation of the */
/* NetWare Requester. Additionally, LAPS statements are needed for the */
/* connection to the server. */
/* ----- */

outline = 'DEVICE=' ComDrv || '\IBMCOM\LANMSGDD.OS2 /I:' || ComDrv || '\IBMCOM'
call lineout CltDrv || '\config.sys', outline
outline = 'DEVICE=' ComDrv || '\IBMCOM\PROTMAN.OS2 /I:' || ComDrv || '\IBMCOM'
call lineout CltDrv || '\config.sys', outline
call lineout CltDrv || '\config.sys', ' '
call lineout CltDrv || '\config.sys', 'REM'
call lineout CltDrv || '\config.sys', 'REM Beginning of NetWare device statements'
call lineout CltDrv || '\config.sys', 'REM'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\LSL.SYS'
call lineout CltDrv || '\config.sys', 'RUN=' NWDrv || '\NETWARE\DDAEMON.EXE'
call lineout CltDrv || '\config.sys', 'DEVICE=' ComDrv || '\IBMCOM\PROTOCOL\ODI2NDI.OS2'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\ROUTE.SYS'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\IPX.SYS'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\NWREQ.SYS'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\SPX.SYS'
call lineout CltDrv || '\config.sys', 'RUN=' NWDrv || '\NETWARE\SPDAEMON.EXE'
call lineout CltDrv || '\config.sys', 'IFS=' NWDrv || '\NETWARE\NWIFS.IFS'
call lineout CltDrv || '\config.sys', 'RUN=' NWDrv || '\NETWARE\NWDAEMON.EXE'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\VIPX.SYS'
call lineout CltDrv || '\config.sys', 'DEVICE=' NWDrv || '\NETWARE\VSHELL.SYS'
call lineout CltDrv || '\config.sys', 'REM'
call lineout CltDrv || '\config.sys', 'REM - NetWare Requester statements END --'
call lineout CltDrv || '\config.sys'

/* ----- */
/* Add LAPS statements */
/* ----- */

call lineout CltDrv || '\config.sys', 'RUN=' ComDrv || '\IBMCOM\PROTOCOL\NETBIND.EXE'
call lineout CltDrv || '\config.sys', 'RUN=' ComDrv || '\IBMCOM\LANMSGEX.EXE'
call lineout CltDrv || '\config.sys', 'device=' ComDrv || '\IBMCOM\PROTOCOL\NETBEUI.OS2'
call lineout CltDrv || '\config.sys', 'device=' ComDrv || '\IBMCOM\PROTOCOL\NETBIOS.OS2'
call lineout CltDrv || '\config.sys', 'device=' ComDrv || '\IBMCOM\MACS\IBMTOK.OS2'

```

Figure 30 (Part 2 of 2). NWPREP.CMD Procedure

Before the installation of the client workstation can start, user permissions and mapping statements have to be established on the NetWare server. In order to allow logging that occurs during the installation on the code server, two different permissions and mappings are needed. The client has to get READ and FILE SCAN permissions for the CID directory and CREATE, READ, WRITE, MODIFY and FILE SCAN permissions for the CIDLOG subdirectory, where the installation logs of the client are found. In the LOGIN script for the client workstation the following statement should be added to provide the client with the necessary code to execute the mappings.

LOGIN Script**MAP L:=SYS:PUBLIC:**

The STARTRFI.CMD explained in 12.4.2, “Adding LAN Transport System to Client diskettes” on page 322 supplies the mapping statements for these two directories where the drive letters X: for the CID directory and L: for the CIDLOG directory are assigned. The initial mapping of L: is changed to the standard LCU log directory assignment. These mappings are used to be consistent with the standard LCU installs. If you want to change these mappings to other drive letters be aware that there are several procedures that have to be changed accordingly: all procedures named in this chapter and all product definitions in the default LCU command file.

In 4.4.7.4, “NetWare LCU Command File” on page 168 the installation part of the default LCU command file provided in the NetWare subdirectory of the sample code CDROM is shown. This command file can be used for initial installs of a client workstation using NetWare as LAN transport system. Additional products can be added.

4.3 Handling of Locked Files

During a product installation, it is possible that certain files that are to be replaced by the installation procedure are already in use by another application running on the client. In this case, the files are *locked* by the operating system and cannot be directly replaced. This problem is related to the case, that several parts of products may be included in different products, for example User Profile Management and First Failure Support Technology/2 are part of LAN Server V5.0, Extended Services 1.0, CM/2 V1.1 and DB2/2 V2.11, CM Server V4.0, PC/3270 for OS/2 V4.1.

In a CID environment, this is a condition that needs to be accounted for. This is because the installation may be initiated by an administrator or software distribution manager that is not aware of the current state of the target system. Even if they are aware, there may be no way to avoid dealing with files that are locked by the operating system. Therefore, the installation program has to find a way to handle the locked files.

Important

The installation programs of CID-enabled products do have a method of handling locked files. Thus, there is no need of any additional activity by the administrator.

This chapter will discuss one method of handling locked files that was developed by IBM. LAN Server V3.0, Extended Services 1.0 and its follow-on products share many common functions and as a result use many of the same files. The solution outlined here was specifically designed for these products, but it provides a model that could be used for developers to design a more generic solution.

This chapter gives the administrator a detailed view on how the locked files problem is handled by the CID-enabled products. As this is additional information that is not part of the necessary knowledge to provide installations you might skip this chapter.

4.3.1 Locked File Solution Using IBMLANLK.EXE

In the following part, we use the example of the LAN Server V3.0 install process to describe how the IBM products named above handle the locked files.

If during the install or remove process, a file is unable to be replaced or deleted, the following will be done:

- If deleting the file, then the name of the file will be saved along with an indication that it is to be deleted. This information is written to the file OS2INSTALLIBMLANLK.LST.
- If trying to replace a file, then the file will be saved under a temporary directory (IBMLANLK). The subdirectory structure under IBMLANLK that the file is saved in will be the same as the subdirectory structure where the file is to be replaced. For example if the file SAMPLE.FIL was supposed to be copied to the D:IBMLANNETPROG subdirectory, then it would be copied to the D:IBMLANLKIBMLANNETPROG subdirectory. For every logical drive where locked files need to be replaced, the temporary subdirectory (IBMLANLK) will be created.

The name of the file placed in the temporary subdirectory will also be written to OS2INSTALLIBMLANLK.LST.

- At the end of the installation process, a device driver statement is added as the first device driver in CONFIG.SYS. The statement added is:
DEVICE=X:OS2INSTALLIBMLANLK.SYS X:OS2INSTALLIBMLANLK.LST

where X is the OS/2 boot drive. The next time the system is IPLed, this device driver will be initialized and carry out its specialized function.

The parameter passed to the device driver is the name of the locked file list generated by the installation program. LAN Server V3.0 will use the above name; however, any name is acceptable to the locked file device driver.

The locked file device driver (during its initialization phase) will delete any files that are in the list and copy the files from the temporary subdirectories to the subdirectories where they should have been installed. The locked file device driver runs prior to any Local Security for 386 HPFS being activated and before loading the LAN system (therefore the files are not currently locked while the locked file device driver is running).

Once the initialization phase is complete, the IPL is allowed to continue. The main part of the locked file device driver performs no additional functions.

The next time the system is IPLed, the locked file device driver will not be loaded. There is no requirement for this second IPL to take place in any specific timeframe, since the locked file device driver has no on-going function and will not affect the operation of the system. It is during the next IPL, that the references to the locked file device driver will be removed from CONFIG.SYS and other cleanup functions performed.

By designing the locked file device driver in this manner, the locked file problem can be resolved with a single re-IPL.

4.3.1.1 Additional Information on IBMLANLK.EXE

The locked file device driver is also used to install code that cannot be installed while running the main installation/configuration program. Code like User Profile Management is installed in this manner. The User Profile Management code may be in use by Extended Services 1.0 or even by the installation/configuration program. It should not be deleted or replaced while in use. During installation of LAN Server V3.0 all of the User Profile Management code is unpacked under the subdirectory IBMLANLK. Within the IBMLANLK subdirectory it will be in the same structure as if installing User Profile Management in its permanent location.

The locked file device driver then installs the User Profile Management code on the re-IPL of the workstation. All code that is common to Extended Services 1.0 and LAN Server V3.0 is treated in this manner. Also code like the installation/configuration program itself is treated this way.

When the locked file device driver runs, a program called **IBMLANLK.EXE** is also executed after the locked file device driver has completed. The IBMLANLK.EXE program is started by the following RUN statement in CONFIG.SYS:

```
RUN=X:OS2INSTALLIBMLANLK.EXE X:OS2INSTALLIBMLANLK.LST
```

where X: is the OS/2 boot drive and the parameter is the name of the locked file list generated by the main installation/configuration program. This statement is added to CONFIG.SYS at the same time as the device driver statement.

The IBMLANLK.EXE program cleans up any requests that the locked file device driver is unable to handle. The device driver is unable to delete subdirectories and this is done by the IBMLANLK.EXE program. In addition to the above, IBMLANLK.EXE is capable of doing a DosExecPgm based on the contents of the IBMLANLK.LST file. This is used to run programs which must be executed after the IPL.

The IBMLANLK.EXE program also removes the locked file device driver and RUN= statements from CONFIG.SYS. This step is accomplished on the second IPL.

Since the locked file device driver and the IBMLANLK.EXE are responsible for the final deletion of files during a removal, they will remain on the hard file.

The following commands are legal in the IBMLANLK.LST file:

DEL	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
DELETE	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
ERASE	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
MOVE	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
COPY	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
REN	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
RENAME	(processed by Locked File DD if possible, else attempted by IBMLANLK.EXE)
RMDIR	(processed by IBMLANLK.EXE)
RD	(processed by IBMLANLK.EXE)
MKDIR	(processed by IBMLANLK.EXE)
MD	(processed by IBMLANLK.EXE)
DOSX	(this is a non-DOS function and results in a DosExecPgm being done). It is executed only by IBMLANLK.EXE.
RMTREE	(this removes the subdirectory and all subdirectories and files under the subdirectory). It is executed only by IBMLANLK.EXE. This command will not be executed if Local Security for 386 HPFS is in the process of being set up (i.e. SETSECUR is in STARTUP.CMD). SETSECUR causes a reboot and the RMTREE will be executed after Local Security for 386 HPFS has been set up.

The following is an example of how IBMLANLK.LST looks when reinstalling the LAN server.

Each command (that is, MOVE) will be a single line in IBMLANLK.LST.

```
RMTREE C:\IBMLANLK
MOVE C:\IBMLANLK\IBMLAN\SERVICES\WKSTAHLP.EXE C:\IBMLAN\SERVICES\WKSTAHLP.EXE
MOVE C:\IBMLANLK\IBMLAN\SERVICES\LSCLIENT.EXE C:\IBMLAN\SERVICES\LSCLIENT.EXE
MOVE C:\IBMLANLK\IBMLAN\NETLIB\LRSE.DLL C:\IBMLAN\NETLIB\LRSE.DLL
MOVE C:\IBMLANLK\IBMLAN\NETLIB\LRNO.DLL C:\IBMLAN\NETLIB\LRNO.DLL
MOVE C:\IBMLANLK\IBMLAN\NETLIB\LRSD.DLL C:\IBMLAN\NETLIB\LRSD.DLL
MOVE C:\IBMLANLK\IBMLAN\NETLIB\LRRS.DLL C:\IBMLAN\NETLIB\LRRS.DLL
MOVE C:\IBMLANLK\IBMLAN\SERVICES\WKSTA.EXE C:\IBMLAN\SERVICES\WKSTA.EXE
MOVE C:\IBMLANLK\IBMLAN\SERVICES\MSRV.EXE C:\IBMLAN\SERVICES\MSRV.EXE
MOVE C:\IBMLANLK\IBMLAN\SERVICES\NETPOPUP.EXE C:\IBMLAN\SERVICES\NETPOPUP.EXE
DOSX C:\IBMLAN\NETPROG\ADDSVRIN.EXE LANCESRV 2 C:\IBMLAN
RMTREE C:\IBMLAN\INSTALL\IBMLAN\INSTALL\IBMCOM
RMTREE C:\IBMLAN\INSTALL\IBMLAN
RMDIR C:\IBMLAN\INSTALL
```

Note

IBMLANLK.LST is processed from top to bottom by the locked file device driver. Any commands that it is capable of executing are executed and removed from the list. The IBMLANLK.LST must have an end-of-file mark in order to be processed correctly. Next the file is processed from top to bottom by IBMLANLK.EXE to clean up any commands that the locked file device driver was unable to process.

This is why, in the above example, it was OK to do the

```
RMTREE C:\IBMLANLK
```

prior to specifying the MOVEs.

4.3.2 Locked File Handling Using NetView DM/2 V2.1

Using NetView DM/2 V2.1 as the software distribution manager, there is the possibility to use the NetView DM/2 V2.1 functions to handle locked files for products that are not-CID enabled. NetView DM/2 V2.1 offers an installation to a so-called Service Area, a temporary file that is moved to its defined target directory, during the next reboot. The function of the NetView DM/2 V2.1 locked file driver is very similar to the IBMLANLK locked file device driver. If you want to use this function, Install to Service Area has to be specified in the PM panels when preparing the installation, or, if you are using line commands, the CDM INSTALL command has to include the parameter /DA:S. Be aware, that you need an ACTIVATE of the client after the install to the Service Area before the changes take effect.

Important

If you are installing CID-enabled products that are capable of handling locked files by themselves, for example with the IBMLANLK.EXE, there is no need to use the locked files solution provided by NetView DM/2 V2.1. Generally, the CID-enabled products do handle the locked files on their own.

4.4 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.

You will find a CONNECT.CMD in the SAMPLECONNECT subdirectory. CONNECT.CMD is a skeleton file that can be altered as required by the administrator. These command files are prepared for a large number of installations. As you will not use all of the product definitions you may cut out those that are not needed. Each installable product gets an index number. These index numbers are generated by the counter variable 'i'. Please remember 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.

On the sample code CDROM there are three different CONNECT.CMDs; the one copied to your system is for the chosen type of server. For examples of the various LCU command files used, refer to:

- 4.4.7.1, "MPTS SRVIFS LCU Command File" on page 163 for SRVIFS LCU
- 4.4.7.2, "LAN Server V5.0 RIPL LCU Command File" on page 164 for RIPL
- 4.4.7.3, "TCP/IP LCU Command File" on page 167 for TCP/IP

Attention!

To allow for the storage of different versions of OS/2 under the CID directory structure the paths to the executable files and to the DLL subdirectory have changed. Please reflect these changes when using LCU command files from sources other than the sample code CDROM of this document.

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 LAN Transport System diskettes, then a lightly attended installation will take place, and an unattended installation when started from a client's hard disk with an OS/2 operating system already installed.

4.4.1 LCU Overview

Packaged with the IBM Multi-Protocol Transport Services product, IBM is shipping a utility called LCU. This utility is designed to allow an administrator to easily chain together a series of CID installs. For example, an end user system may require OS/2, CM/2 and LAPS from IBM Multi-Protocol Transport Services to be installed. Though each product is individually enabled for CID, there is the obvious requirement to allow the complete install of all these products to be invoked as a single process instead of several processes. LCU tracks the current state of installation across IPLs and ensures that each CID install program executes in the correct sequence. Once the administrator has defined the desired sequence to LCU, the installation process will run to completion without end user involvement at the client workstation. However, an end user must always be at the client workstation to do one of the following:

- Insert the two diskettes created on the server and reboot
- or
- Enable the client workstation to talk to the server and reboot

This is called lightly attended installation; please refer to 1.3, "Installation Modes" on page 20 for a complete description of the different types of installations in a CID environment.

4.4.2 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:

1. Successful completion, reboot not required
2. Successful completion, reboot
3. Successful completion, reboot and call me back
4. Error

If you later need more information see K.4, “Architected CID Return Codes” on page 602.

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 via the REMOTE_INSTALL_STATE state variable.

4.4.2.1 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 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.

When using LAN Server V5.0, NetWare or TCP/IP V3.0 the LCU command file is called directly in `STARTUP.CMD` without using the LCU agent in the examples in this book.

- The saved state variable is interrogated by LCU to detect infinite loops and for product install programs to determine their execution state.

4.4.2.2 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 and 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 some other products without state-variable too. It may cause you some problems, so it can be easier to install a program that requires a reboot in a separate section.

4.4.3 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.

4.4.3.1 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
3. 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 LTS 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.

4.4.3.2 Default Response File

LCU can also do automatic default response file selection. See "Default Response File" on page 150 for a detailed discussion on this subject. 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 an LTS diskette with a correctly coded client name.

4.4.4 LCU Command File Structure

For a detailed listing of the LCU DEFAULT command file, please refer to the CIDCLIENTCONNECT.CMD. This is the file used in all examples in this book.

With MPTS LCU three sample LCU command files are provided:

- CASSAMP1.CMD includes example of Service Pak installation
- CASSAMP2.CMD includes example of Service Pak installation
- CASSKEL.CMD skeleton file to be used for modifications

The LCU REXX command file is composed of 3 basic sections; the following sections describe each of the 3 command file sections.

4.4.4.1 First Section of LCU Command File

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:' ← Replace with the drive which the client will be booted from
configsys = bootdrive || '\CONFIG.SYS'
exepath = 'X:\EXE\OS2V300' ← Replace with the path where the SETBOOT.EXE is located
```

Please take care to ensure that the exepath really points to the OS/2 version that will be installed on the client. (Or the OS/2 version that is installed if only other products will be installed with the LCU command file.)

Product Data Section: The following statements are product specific data. Each product, which will be installed, needs a set of these statements. The program specific parameters are linked together via the "comma" at the end of each statement. This example is for OS/2 operating system install SEINST.

```
x.seinst211 = 2
x.2.name='OS/2 2.11'
x.2.statevar = 'CAS_' || x.2.name
x.2.instprog = 'x:\exe\os2v211\seinst',
              '/b:' || bootdrive || ',
PROGRAM      '/s:x:\img\os2v211',
SPECIFIC     '/t:c:\service',
PARAMETERS   '/l1:L:\os2v211\' || client || '.log ',
              '/r:'
x.2.rspdir   = 'x:\rsp\os2v211'
x.2.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'
x.i.statevar = 'CAS_' || x.i.name
x.i.instprog = 'x:\img\MPTS\MPTS',
              '/e:maint',
              '/s:x:\img\MPTS',
              '/t:' || bootdrive || '\',
              '/l1:L:\MPTS\' || client || '.log ',
              '/r:'
x.i.rspdir   = 'x:\rsp\MPTS'
x.i.default  = 'MPTS.RSP'
```

The following table describes the product variables.

<i>Table 8. Product Variable Descriptions</i>		
Variable	Title	Description
x.seinst211	Structure index	This contains the name of the install program and a number to identify the program. This example is for SEINST installing OS/2 V2.11 operating system.
x.1.name	Product name	A user defined name for this product, for example OS/2 2.11. This name must be unique for each of the install programs, it is used for messages and building the value for x.1.statevar.
x.1.statevar	State variable name	<p>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.</p> <p>NOTE: The statevar keyword must always be defined. If a state variable is not specified in the product data section for a program, that program will run any time the LCU REXX command file encounters it. Not specified is indicated by a NULL string "" example: x.1.statevar=". If there is any chance that a program would request to be called back, a state variable MUST be specified.</p>
x.1.instprog	Fully qualified install program name	The name of the install program for this product with its path and specific parameters.
x.1.rspdir	Response file directory	The path to the response files for this product.
x.1.default	Default response file name	The name of the default response file to be used if the one for this client cannot be found. Response files in LCU may have the name <client name>.RSP.

Default Response File: The LCU command file can do automatic default response file selection. The program will check the directory specified in x.1.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.1.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.seinst211 = 2
x.2.name='OS/2 2.11'
x.2.statevar = 'CAS_' || x.2.name
x.2.instprog = 'x:\exe\os2v211\seinst          ',
               '/b:' || bootdrive || '      ',
               '/s:x:\img\os2v211          ',
               '/t:c:\service              ',
               '/l2:L:\os2v211\' || client || '.log ',
               '/r:'
x.2.rspdir   = 'x:\rsp\os2v211'
x.2.default = 'default.rsp'
```

If you wish to hard code a specific response file, you must set `"rspdir"` and `"default"` to `''`. (`''` indicates a NULL string).

```
x.seinst21 = 1
x.1.name='OS2V21'
x.1.statevar = 'CAS_' || x.1.name
x.1.instprog = 'x:\exe\seinst          ',
               '/b:' || bootdrive || '      ',
               '/s:x:\img\os2v21          ',
               '/t:c:\service              ',
               '/l1:L:\os2v21\' || client || '.log ',
               '/r:specific.rsp'
x.1.rspdir   = ''
x.1.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 = 49
```

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: SERVER1ALIAS'  
'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. One drive alias could be located on another server and used for backup purposes. Client workstations could be backed up to the other server before starting the OS/2 installation to minimize the load on the code server.

Sample CONNECT.CMD

In the sample CONNECT.CMDs provided with this book in the product data section there are product variables for many of the current IBM programs and versions of these programs.

For each product variable a state variable will be created and written to the client's CONFIG.SYS during installation. This slows down the installation process unnecessarily. Use the CONNECT.CMD as a template and remove those product variables that will not be used and create your own 'default' for those products used within your environment.

4.4.4.2 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:

```

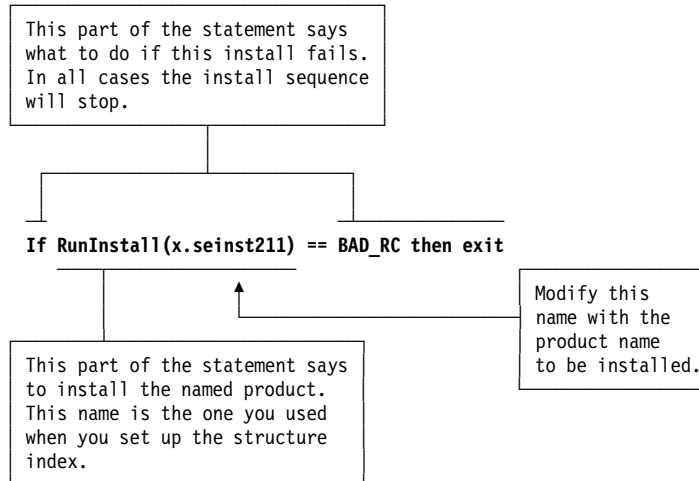
Do Forever
Select
  when OVERALL_STATE = 0 then do
    if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                              /* if it was, then goto state 1*/
    if RunInstall(x.semaint) == BAD_RC then exit /* Install maintenance system */
    if RunInstall(x.MPTS_prep) == BAD_RC then exit /* Install MPTS prep system */
    if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.casin1) == BAD_RC then exit /* Install LCU */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 1 then do
    if RunInstall(x.CONNECT) == BAD_RC then exit /* Install operating system */
    if RunInstall(x.MPTS) == BAD_RC then exit /* Install MPTS */
    if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.casin1) == BAD_RC then exit /* Install LCU */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 2 then do
    'SET CMWAIT=1'
    if RunInstall(x.CM2) == BAD_RC then exit /* Install CM/2 */
    Call CheckBoot
  end
  when OVERALL_STATE = 3 then do
    if RunInstall(x.laninstr) == BAD_RC then exit /* Install LAN Server 5.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 4 then do
    if RunInstall(x.TCPIP30) == BAD_RC then exit /* Install TCP/IP Version 3.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 5 then do
    if RunInstall(x.ifsdel) == BAD_RC then exit /* Delete SRVIFS requester */
    if RunInstall(x.casdelete) == BAD_RC then exit /* Delete LCU */
    Call Reboot /* Reboot */
  end
end
end
exit

```

Following is a definition of the various lines in section two:

Statement	Description
Select	REXX function name
When	REXX instruction used to determine what should be run after each reboot
RunInstall	The command to install a product

Description of the **RunInstall** statement:



Note on RunInstall

The last two RunInstall statements are "IFSDEL" and "CASDELETE"; these statements remove THINIFS (LCU redirector) and LCU. You should NOT remove these statements since doing so will cause the final reboot to reinitiate the install process.

4.4.4.3 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.

4.4.5 Adding Products to the LCU Command File

Any CID enabled product and some non-CID enabled products can be installed with LCU. The administrator must do the following modifications to the LCU command file:

- Add a set of product specific data in the product data section.
- Increment the NUM_INSTALL_PROGS variable.
- Add a "when OVERALL_STATE...." function to the install sequence.
- Include RunInstall and Checkboot statements.
- Adjust "OVERALL_STATE = ..." statements to be in sequence.

Note on adding products to the LCU command file

The "when OVERALL_STATE...." function that contains "IFSDEL" and "CASDELETE" MUST be kept last. Insert your new "when OVERALL_STATE...." function ahead of it and adjust the "OVERALL_STATE = ..." numbers to be sequential.

You have to put the images or files of the product in the IMG<PRODUCTNAME> subdirectory of the code server and point to this directory in the product description. If the product is CID enabled, you have to create a proper response file and put it in the RSP<PRODUCTNAME> subdirectory. And do not forget to create the a subdirectory for log files LOG<PRODUCTNAME>.

Attention LAN Server administrators

After the creation of the new directories do not forget to **apply** the access control profiles for

- CIDIMG
- CIDRSP
- CIDLOG

You can not do it with one command for the whole CID structure, since the clients need the additional WRITE access right to the LOG directories.

For example if you want to add hard disk preparation prior to installation and the Remote Multiple Printer Installation Application (RMPI) see below. For more information refer to Chapter 8, "Auto-Partitioning the Hard Disk" on page 243 and Chapter 7, "Remote Multiple Printer Support" on page 217.

The following example shows extensions of the Do Forever Loop adding hard disk preparation and the Remote Multiple Printer Installation Application.

```

Do Forever
Select
  when OVERALL_STATE = 0 then do
    if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                              /* if it was, then goto state 1*/
    if RunInstall(x.semaint) == BAD_RC then exit /* Install maintenance system */
    if RunInstall(x.MPTS_prep) == BAD_RC then exit /* Install MPTS prep system */
    if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 1 then do
    if RunInstall(x.diskprp) == BAD_RC then exit /* Prepare hard drive */
    if RunInstall(x.CONNECT) == BAD_RC then exit /* Install operating system */
    if RunInstall(x.MPTS) == BAD_RC then exit /* Install MPTS */
    if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 2 then do
    'SET CMWAIT=1'
    if RunInstall(x.CM2) == BAD_RC then exit /* Install Extended Services */
    Call CheckBoot
  end
  when OVERALL_STATE = 3 then do
    if RunInstall(x.laninstr) == BAD_RC then exit /* Install LAN Server 5.0 */
    if RunInstall(x.rinstprn) == BAD_RC then exit /* Install Remote Printers */
    Call CheckBoot
  end
  when OVERALL_STATE = 4 then do
    if RunInstall(x.TCPIP30) == BAD_RC then exit /* Install TCP/IP Version 3.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 5 then do
    if RunInstall(x.ifsdel) == BAD_RC then exit /* Delete SRVIFS requester */
    if RunInstall(x.casdelet) == BAD_RC then exit /* Delete LCU */
    Call Reboot /* Reboot */
  end
end
end
exit

```

4.4.6 LCU Command File Execution of a Diskette Initiated Installation

This section will describe the LCU command file execution flow. The following is a walk-through of the installation of an OS/2 operating system on a diskette-initiated system.

The following figure describes the statements needed for the installation of OS/2 base operating system on a diskette-initiated system.

	RUN #1	RUN #2	RUN #3	
	-----	-----	-----	
Do Forever				
Select				
when OVERALL_STATE = 0 then do	1			
if BootDrive() == 'DISKETTE' then iterate	2			
if RunInstall(x.semaint) == BAD_RC then exit				
if RunInstall(x.mpts_prep) == BAD_RC then exit				
if RunInstall(x.thinifs1) == BAD_RC then exit				
if RunInstall(x.thinifs2) == BAD_RC then exit				
if RunInstall(x.casinstl) == BAD_RC then exit				
Call CheckBoot				
end				
when OVERALL_STATE = 1 then do	3	10		
if RunInstall(x.connect) == BAD_RC then exit	4	11		
if RunInstall(x.mpts) == BAD_RC then exit	5	12		
if RunInstall(x.thinifs1) == BAD_RC then exit	6	13		
if RunInstall(x.thinifs2) == BAD_RC then exit	7	14		
if RunInstall(x.casinstl) == BAD_RC then exit	8	15		
Call CheckBoot	9	16		
end				
when OVERALL_STATE = 2 then do			17	
if RunInstall(x.ifsdel) == BAD_RC then exit			18	
if RunInstall(x.casdelet) == BAD_RC then exit			19	
Call Reboot			20	
end				
end				
end				
exit				

The following is the sequence in which the statements are executed. There are 20 different steps required for a successful completion of this scenario.

Please note that all statements between "OVERALL_STATE=.." and the corresponding "end" are part of the same queue.

QUEUE1 = bootdrive+semaint+mpts_prep+thinifs1+thinifs2+casinstl

QUEUE2 = connect+mpts+thinifs1+thinifs2+casinstl

QUEUE3 = ifsdel+casdelet

The numbers to the right of the installation statements correspond to the numbers in the detailed explanation section that follows.

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. All statements between this statement and the corresponding "end" are part of the same queue named QUEUE1.

2. if BootDrive() == "DISKETTE" then iterate

This statement will check if the boot drive is removable. If the drive booted from is a diskette drive, then the OVERALL_STATE is set to OVERALL_STATE+1. If the installation was started from a boot diskette, then the LCU command file will skip QUEUE1 and execute the statements in QUEUE2.

This test will also be true and QUEUE1 will be skipped when the client is RIPLed from a LAN Server V5.0.

3. when OVERALL_STATE = 1 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 1. All statements between this statement and the corresponding "end" are part of the same queue named QUEUE2.

4. if RunInstall(x.connect) == 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 callback by using return code x'FF02' because the installation was booted from diskette.

5. if RunInstall(x.mpts) == BAD_RC then exit

This statement will install MPTS for the production system. The boot drive CONFIG.SYS is modified in this step. This program will request a reboot and will not request to be called back.

6. 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 default 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=targetSRVIFS.SYS**
- **IFS=targetSRVIFSC.IFS <options>**
- **CALL=targetSRVATTCH.EXE drive_letter: servername**

In addition, the PATH statement is also updated to include the target of the installation. This program will request a reboot. Please refer to 4.1.3.1, "THINIFS" on page 94 for a description of the THINIFS parameters.

7. if RunInstall(x.thinifs2) == BAD_RC then exit

This statement will install the LCU redirector again in the same QUEUE. THINIFS executes twice in the same queue in order to attach to a LOG redirected drive called L: 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 statements are added to the CONFIG.SYS:

- **DEVICE=targetSRVIFS.SYS**
- **IFS=targetSRVIFSC.IFS <options>**
- **CALL=targetSRVATTCH.EXE drive_letter: servernamealias**

8. if RunInstall(x.casinstl) == 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.

9. 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 callback. The OVERALL_STATE variable CAS_STATE is set to 1 so that when the workstation is rebooted the LCU command file will enter the same state again and re-execute QUEUE2. SEINST asked to be called back by issuing return code x'FF02'; therefore LCU is setting its state variable CAS_OS/2 2.11=2, so that after the reboot SEINST knows that it is entering this state because it asked to be called back.

The following figure shows the modifications of CONFIG.SYS at the time of the first reboot.

LCU command file execution of a diskette-initiated installation

This CONFIG.SYS is an intermediate CONFIG.SYS that will never be seen by the end user if no error occur during execution of the LCU command file. The purpose of the figure is to explain the reboot mechanism involved when LCU executes a particular sequence of program installs. Modifications to the CONFIG.SYS are highlighted in this figure and unchanged lines are excluded and the complete paths are not shown.

```

.
LIBPATH=C:\IBMCOM\DLL;.;C:\OS2\DLL;C:\OS2\MDOS;C:\; ...
... C:\OS2\APPS\DLL;X:\DLL\CONNECT;X:\IMG\LCU;
SET PATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\MDOS\WINOS2; ...
... C:\OS2\INSTALL;C:\;C:\OS2\MDOS;C:\OS2\APPS;C:\SR
VIFSRQ;
SET DPATH=C:\IBMCOM;C:\OS2;C:\OS2\SYSTEM;C:\OS2\MDOS\WINOS2; ...
... C:\OS2\INSTALL;C:\;C:\OS2\BITMAP;
... C:\OS2\MDOS;C:\OS2\APPS;
.
DEVICE=c:\IBMCOM\PROTOCOL\LANPDD.OS2
DEVICE=c:\IBMCOM\PROTOCOL\LANVDD.OS2
DEVICE=c:\IBMCOM\LANMSGDD.OS2 /I:c:\IBMCOM
DEVICE=c:\IBMCOM\PROTMAN.OS2 /I:c:\IBMCOM
.
.
RUN=c:\IBMCOM\PROTOCOL\NETBIND.EXE
RUN=c:\IBMCOM\LANMSGEX.EXE
DEVICE=c:\IBMCOM\PROTOCOL\NETBEUI.OS2
DEVICE=c:\IBMCOM\PROTOCOL\NETBIOS.OS2
DEVICE=c:\IBMCOM\PROTOCOL\LANDD.OS2
DEVICE=c:\IBMCOM\PROTOCOL\LANDLLDD.OS2
DEVICE=c:\IBMCOM\MACS\IBMTOK.OS2
RUN=c:\IBMCOM\PROTOCOL\LANDLL.EXE
CALL=c:\srvfifsrq\SRVATTCH.EXE x: CLIENT1
DEVICE=c:\srvfifsrq\SRVIFS.SYS
IFS=c:\srvfifsrq\SRVIFSC.IFS CLIENT1
CALL=c:\srvfifsrq\SRVATTCH.EXE L: \\CIDSrv\LOG
RUN=X:\IMG\LCU\SRVREXX.EXE
SET CAS_STATE=1
SET CAS_OS/2 WARP CONNECT=2
SET CAS_OS/2 WARP CONNECT MAINTENANCE=0
SET CAS_MPTS MAINTENANCE INSTALLATION=0
SET CAS_MPTS =0
SET CAS_LS 50 Requester=0
SET CAS_CM Server for OS/2=0
SET CAS_CM/2 1.11=0
SET CAS_PC/3270 for OS/2 Version 4.1=0
SET CAS_TCP/IP 3.0 for OS/2=0
SET CAS_SRVIFS SERVER=0
.
.

```

Figure 31. Modifications in CONFIG.SYS at First Reboot

10. when OVERALL_STATE = 1 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 1. All statements between this statement and the corresponding "end" are part of the same queue named QUEUE2. We are entering this state again and re-execute QUEUE2 because SEINST requested to be called back.

11. 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_OS/2 2.11=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.

12. 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 on RunInstall(x.mpts)

Remember that programs having a state variable defined will never run again the second time the LCU command file encounters them.

13. if RunInstall(x.thinifs1) == BAD_RC then exit

The second time through this state, this program will install LCU redirector 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. This program will request a reboot.

14. if RunInstall(x.thinifs2) == BAD_RC then exit

The second time through this state, this program will install LCU redirector 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. This program will request a reboot.

15. if RunInstall(x.casinstl) == BAD_RC then exit

The second time through this state, this program will install LCU 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.

16. 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 2 so that when the workstation is rebooted the LCU command file will enter in the next state "OVERALL_STATE=2" and now execute QUEUE3.

17. when OVERALL_STATE = 2 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 2.

18. 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. Pathing statements from the PATH, DPATH or LIBPATH will not be removed. This program will request a reboot.

19. 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.

20. Call Reboot

This statement will reboot the machine, and is the last reboot. When the machine reboots, OS/2 operating system and MPTS configured for token-ring are successfully installed.

4.4.7 The LCU Command File - Samples and Skeletons

The key section LCU command file is the 'Do Forever Loop' shown below for each type of installation.

4.4.7.1 MPTS SRVIFS LCU Command File

```

Do Forever
Select
  when OVERALL_STATE = 0 then do
    if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                              /* if it was, then goto state 1*/
    if RunInstall(x.semaint) == BAD_RC then exit /* Install maintenance system */
    if RunInstall(x.MPTS_prep) == BAD_RC then exit /* Install MPTS prep system */
    if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 1 then do
    if RunInstall(x.CONNECT) == BAD_RC then exit /* Install operating system */
    if RunInstall(x.MPTS) == BAD_RC then exit /* Install MPTS */
    if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
    if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 2 then do
    if RunInstall(x.PCOMOS2V41) == BAD_RC then exit /* Install PC/3270 for OS/2 4.1 */
    Call CheckBoot
  end
  when OVERALL_STATE = 3 then do
    if RunInstall(x.lanreq) == BAD_RC then exit /* Install LAN Requester V. 5.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 4 then do
    if RunInstall(x.TCPIP30) == BAD_RC then exit /* Install TCP/IP Version 3.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 5 then do
    if RunInstall(x.DB2SU) == BAD_RC then exit /* Install DB2/2 2.11 Single User*/
    Call CheckBoot
  end
  when OVERALL_STATE = 6 then do
    if RunInstall(x.ifsdel) == BAD_RC then exit /* Delete SRVIFS requester */
    if RunInstall(x.casdelet) == BAD_RC then exit /* Delete LCU */
    Call Reboot /* Reboot */
  end
end
end
exit

```

4.4.7.2 LAN Server V5.0 RIPL LCU Command File

In order to execute a normal CID installation it is necessary to create additional installation procedures that provide the reconnection to the server after a reboot during the installation process. It is not possible to install LAN Requester V5.0 in the first installation sequence because LAN Requester V5.0 needs Presentation Manager and other executable files. To reconnect the client with the server after the first installation sequence and then execute the next installation sequence, a temporary LAN requester is created with the help of THINR300.CMD.

These additional procedures are:

- THINR300.CMD
- REQDELE1.CMD
- REQDL300.CMD
- REQUPDAT.CMD
- RMTREE.CMD

They can be found in the RIPL subdirectory of the sample code CDROM and they are copied to the EXEOS2V300 subdirectory during the setup of the code server.

- THINR300.CMD

The THINR300.CMD installs a temporary requester on the client workstation. It uses files and functions of LAN Server V5.0.

Please note that the command procedure updates LIBPATH, PATH and DPATH and these statements must be modified if you are installing another version than OS/2 Warp V3 (or are using subdirectories other than OS2V300).

- REQUPDAT.CMD

This procedure is executed after the LAN Requester V5.0 installation and changes the value of the LOGONVERIFICATION to DOMAIN. This procedure is OS/2 version independent.

- REQDELE1.CMD

This procedure deletes the CONFIG.SYS statements from the client workstation that were added during the THINR300.CMD procedure. This procedure is OS/2 version independent.

- REQDL300.CMD

This procedure removes the directory trees of the temporary LAN requester. It cleans up the CONFIG.SYS PATH, LIBPATH and DPATH statements of the client for the CID installation process. And needs updating if THINR300.CMD is changed. It also removes the call to the STARTRPL.CMD from the STARTUP.CMD and it deletes the ENV_VARS.CMD file that saved the input from the CRENVVAR.EXE.

- RMTREE.CMD

This procedure is used to delete the whole subdirectory tree of the temporary LAN requester. It is invoked during the REQDL300.CMD. This procedure is also independent of the used OS/2 version.

Below is an excerpt of the sample CONNECT.CMD for CID installations which uses LAN Server V5.0 with RIPL as the code server.

```

Do Forever
  Select
    when OVERALL_STATE = 0 then do
      if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                              /* if it was, then goto state 1*/
      if RunInstall(x.semaint) == BAD_RC then exit /* Install maintenance system */
      if RunInstall(x.MPTS_prep) == BAD_RC then exit /* Install MPTS prep system */
      if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 1 then do
      if RunInstall(x.CONNECT) == BAD_RC then exit /* Install operating system */
      if RunInstall(x.MPTS) == BAD_RC then exit /* Install MPTS */
      if RunInstall(x.thinr300) == BAD_RC then exit /* Install 'thin' LAN req. */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 2 then do
      if RunInstall(x.lanreq) == BAD_RC then exit /* Install LAN Server 5.0 */
      if RunInstall(x.requpdat) == BAD_RC then exit /* Update from thin requester */
      if RunInstall(x.reqdele1) == BAD_RC then exit /* Install Delete first part */
      Call CheckBoot
    end
    when OVERALL_STATE = 3 then do
      if RunInstall(x.PCOMOS2V41) == BAD_RC then exit /* Install PC/3270 for OS/2 4.1 */
      call CheckBoot
    end
    when OVERALL_STATE = 4 then do
      if RunInstall(x.TCPIP30) == BAD_RC then exit /* Install TCP/IP 3.0 */
      Call CheckBoot
    end
    when OVERALL_STATE = 5 then do
      if RunInstall(x.DB2SU) == BAD_RC then exit /* Install DB2/2 2.11 Single User*/
      Call CheckBoot
    end
    when OVERALL_STATE = 6 then do
      if RunInstall(x.reqdl300) == BAD_RC then exit /* Install Delete second part */
      Call Reboot /* Reboot */
    end
  end
end
exit

```

When the real installation of LAN Requester V5.0 is done (in state 2) REQUPDAT is run to ensure that the logon verification is done on the domain. REQDELE1 is run to clean up part of the "thin requester" installed in state 1. In the last state (in this sample state 6) the final cleanup is done with REQDL300.

4.4.7.3 TCP/IP LCU Command File

The following shows the installation part of the default LCU command file for TCP/IP. Executing these installs, a client will have OS/2 Warp Connect, &mpts, TCP/IP V3.0, PC/3270 for OS/2 V4.1, LAN Server V5.0 Requester, and DB2/2 V2.11 Single-User Version Service Pak installed. Please note that the installation order is not optional but determined by product dependencies. See 4.1.10, "Product Installation Order" on page 127 for more information.

```
Do Forever
Select
  when OVERALL_STATE = 0 then do
    if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
    if RunInstall(x.semaint) == BAD_RC then exit /* Install maintenance system */
    if RunInstall(x.tcp_prep) == BAD_RC then exit /* Install TCP/IP client */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 1 then do
    if RunInstall(x.diskprep) == BAD_RC then exit /* Automated HD-Partitionig */
    if RunInstall(x.CONNECT) == BAD_RC then exit /* Install operating system */
    if RunInstall(x.MPTS) == BAD_RC then exit /* Install MPTS */
    if RunInstall(x.thintcp) == BAD_RC then exit /* Install temp. TCP/IP client */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 2 then do
    if RunInstall(x.TCPIP30) == BAD_RC then exit /* Install TCP/IP Version 3.0 */
    if RunInstall(x.tcpcopy) == BAD_RC then exit /* Update STARTUP.CMD */
    Call CheckBoot
  end
  when OVERALL_STATE = 3 then do
    if RunInstall(x.PCOMOS2V41) == BAD_RC then exit /* Install PC/3270 for OS/2 4.1 */
    call CheckBoot
  end
  when OVERALL_STATE = 4 then do
    if RunInstall(x.lanreq) == BAD_RC then exit /* Install LAN Requester V. 5.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 5 then do
    if RunInstall(x.DB2SU) == BAD_RC then exit /* Install DB2/2 2.11 Single User*/
    Call CheckBoot
  end
  when OVERALL_STATE = 6 then do
    if RunInstall(x.tcdelete) == BAD_RC then exit /* Cleanup */
    Call Reboot /* Reboot */
  end
end
end
exit
```

4.4.7.4 NetWare LCU Command File

The following shows the installation part of the default LCU command file for NetWare.

Note on NetWare Environment

The LCU environment for Novell Netware has not been tested again, as this environment is only valid for Netware V.3.12. So in this section there are no modifications made for OS/2 Warp (and Warp Connect).

Executing these installs, a client will have OS/2 V2.11, NetWare requester, LAPS, LAN Server V3.01 requester, DB2/2 V1.0 Single-User Version and the DB2/2 Service Pak installed. Please note that the installation order is not optional but determined by product dependencies. Especially the order of installing LAPS **after** the NetWare requester must be kept to have a working system. See 4.1.10, "Product Installation Order" on page 127 for more information.

```

Do Forever
Select
  when OVERALL_STATE = 0 then do
    if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                              /* if it was, then goto state 1*/
    if RunInstall(x.semaint211) == BAD_RC then exit /* Install maintenance system */
    if RunInstall(x.nwprep) == BAD_RC then exit /* Install NetWare prep system */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 1 then do
    if RunInstall(x.seinst211) == BAD_RC then exit /* Install operating system */
    if RunInstall(x.nwinst) == BAD_RC then exit /* Install NetWare requester */
    if RunInstall(x.laps) == BAD_RC then exit /* Install LAPS */
    Call CheckBoot /* Reboot if it was requested */
  end
  when OVERALL_STATE = 2 then do
    if RunInstall(x.nwicon) == BAD_RC then exit /* Create NetWare Icon */
    if RunInstall(x.lanreqa) == BAD_RC then exit /* Install LAN Requester 3.0 */
    Call CheckBoot
  end
  when OVERALL_STATE = 3 then do
    if RunInstall(x.cm111) == BAD_RC then exit /* Install Comms. Mgr/2 1.11 */
    Call CheckBoot
  end
  when OVERALL_STATE = 4 then do
    if RunInstall(x.db2su10) == BAD_RC then exit /* Install DATABASE 2 OS/2 */
    Call CheckBoot
  end
  when OVERALL_STATE = 5 then do
    if RunInstall(x.wr07015) == BAD_RC then exit /* Install DATABASE 2 OS/2 SP */
    Call CheckBoot
  end
  when OVERALL_STATE = 6 then do
    if RunInstall(x.nwdelete) == BAD_RC then exit /* Clean up for NetWare */
    Call Reboot /* Reboot */
  end
end
end
exit

```

4.5 Using LCU CASPREP Utility

CASPREP is a utility supplied by NTS/2 and MPTS.

For NTS/2 it can be found on the NTS/2 Utilities diskette. It is described completely in the *IBM Network Transport Services/2 Redirected Installation and Configuration Guide*, S96F-8488, Appendix A.

For MPTS it is packed into MPTSAPLT.ZIP found on MPTS diskette 3 in the APPLETS subdirectory. Please refer to the *LAN CID Utility Guide*, S10H-9742 for a complete description on how to unpack and use CASPREP.

The following sections will give a short introduction into CASPREP, but it is mandatory to refer to the manual before using this utility.

CASPREP is a REXX program that will process a script file into an LCU command file. The script file contains keywords, but no REXX syntax. There are two forms of script files delivered with CASPREP:

- Basic Input File with fewer keywords and no default response file.
- Advanced Input File allowing default response file.

CASPREP requires a base file and a user generated file. The base file is shipped with LCU and no modifications are required by the user. CASPREP reads the base and user generated file, meshes the two together and produces an LCU command file.

The following files are shipped with CASPREP:

CASPREP.CMD	The CASPREP utility
CASBASE.FIL	Base command file that the user generated file is integrated with to create the LCU command file
CASADV.FIL	Sample Advanced Input File
CASBASIC.FIL	Sample Basic Input File

CASPREP is invoked with the following syntax:

CASPREP Syntax

```
CASPREP <input.fil> <lcu.cmd> <casbase.fil>
```

The parameters are:

INPUT.FIL	User generated input file
LCU.CMD	LCU.CMD REXX command output file
CASBASE.FIL	Base command file

Before using the NTS/2 versions of the sample files you should update them, because the samples reflect only OS/2 V2.0 but no later versions of either OS/2 or related products.

The MPTS CASBASE.FIL and CASADV.FIL are updated for OS/2 V2.1 and LAN Server V4.0, but needs editing if other OS/2 versions will be installed and to add other products than OS/2, MPTS and LAN Server V4.0.

Note that they also assume a slightly different CID structure than the one suggested in this book. The MPTS sample files need to be changed:

```
from EXEV210 to EXECONNECT  
from DLL\V210 to DLL\CONNECT  
from DISK1\V210 to DISK1\CONNECT
```

otherwise the suggested directory structures are the same.

4.6 NetView DM/2 Change Control Files

In this section we will cover some of the key functions NetView DM/2 provides to perform change management. We will describe in detail the change files needed with NetView DM/2 to perform installs of client workstations. These change files fulfill the same function as the LCU command file used in the LCU environment as they hold all necessary information about the install program. The install sequence, in LCU command files found in the *Do forever* loop, is defined with NetView DM/2 specific commands that group several change files.

For NetView DM/2 V2.1 users the *NetView DM/2 V2.1 CDM User's Guide* Appendixes contain a lot of examples and scenarios.

4.6.1 Objects, Global Names and NetView DM/2 Catalog

Before any software or data can be distributed to a client, it must be prepared. To get recognized by NetView DM/2 it has to be entered as an object to the catalog. The catalog is the local database used by NetView DM/2 to maintain all information needed by the Change Distribution Manager (CDM) component of NetView DM/2 to process objects. Please see the *IBM NetView Distribution Manager/2 Version 2.1 User's Guide*, SH19-5048-02 for more information. How these objects are created will be explained in the next section. First, we want to state how the naming of these objects is organized.

NetView DM/2 allows change management for the client workstations. For now, it is enough to know that change management means that the actual state of what is installed at the client is stored and every change that was done is documented. If you later want to know more about change management please refer to the product documentation. To identify what is installed on the workstations the objects handled by NetView DM/2 do have so-called global names. These global names are defined in the change management process for the SNA/MVS environment to which NetView DM/2 is related with its possible connection to NetView DM/MVS.

SNA/MVS change management defines objects through network-unique global names. This will allow a company wide change management process. It is therefore recommended to spend some time on the naming conventions for NetView DM/2 objects as this will also affect your host environment. The global names used in this book are **not** meant as guidelines for your naming but as simple examples.

The global name consists of 2 - 10 tokens following conventions that are thoroughly described in the *IBM NetView Distribution Manager/2 Version 2.1 User's Guide*, SH19-5048-02. For our purposes they can be summarized: The first 1-7 tokens contain the component name. They are followed by the change type. The change type can be refresh, update or fix (REF, UPD, FIX respectively). The change type is followed by a level number and a version description of one or more tokens. Here is an example for a refresh:

COMPANY.PRODUCT.EXTRAINFO.REF.001.VERSION3.ENGLISH

In NetView DM, all objects have the same global name as in the NetView DM/2 catalog. The objects are called resources in NetView DM/MVS. There are several types of objects architected in SNA/MVS. They are listed below together with the abbreviations used by NetView DM/2:

- Flat data objects (FLATData)
This type is a single data file and not a change file.
- Maintenance information objects (DUMP, CONFIGfile, TRACE, ERRLOG)
These are also single flat files.
- Relational data objects (RELData)
The relational data object is a special case of flat data being an exported database table.
- OS/2 procedures (PROCEDURE)
Procedure objects are OS/2 command procedures or executables.
- Change management objects (SOFTWARE, MICRocode, FLATData)
Microcode and software are synonymous in NetView DM/2. Software is the object type used in this publication. A change management object is a package that can contain more than one file. For example, flat data files or procedures may be part of a change file.

A NetView DM/2 catalog entry consists of the global name of the object and the local file, including some information about compression. For change management objects this local file is always a change file.

4.6.2 Change Files and Change File Profiles

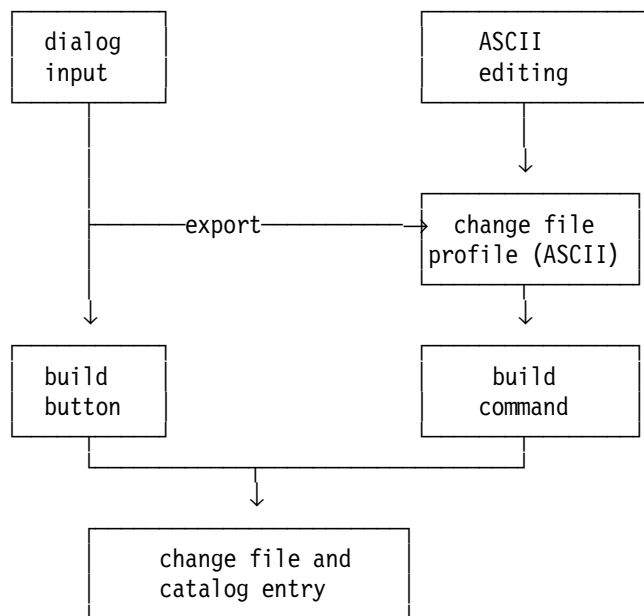
Installation of applications through NetView DM/2 is accomplished via change files. A change file has to be reflected by a catalog entry to be an object that can be distributed. A change file may contain an instruction to execute a program or command on the CC client and/or a list of files to be distributed to the CC client.

The change file cannot be edited directly. To build the change file, specifications about files and the installation program are entered through the dialog interface or into an ASCII file. This flat ASCII file is called the **change file profile**.

The dialog interface allows entering the necessary information in PM panels which is then transformed by the NetView DM/2 CDM BUILD command directly into a change file. The NetView DM/2 CDM BUILD command does the same by having a change file profile as input.

Instead of typing all keywords into an ASCII file, the dialog interface input can optionally be saved by using the export option. This will generate a valid change file profile. An exported change file profile can be used later as is, or modified by an editor.

In both cases the output consists of a change file and a catalog entry. The following picture illustrates the above:



The ASCII change file profile consists of four parts:

- **Global Statements**

They contain the target directory (TargetDir keyword) and the length of the component name (CompNameLen keyword). CompNameLen is an optional keyword and defines how many of the tokens in the component part of the global name are used to define the installation target.

- **Catalog Section**

Most of the catalog entry information is put here. The three keywords are ObjectType (SOFTWare, FLATData or MICRocode), GlobalName and Description (free text). The catalog is scanned for entries with the same global name tokens because the global name has to be unique.

The path of the change file will be specified as an argument in the Build command. See 4.6.3.1, “Change Files from Profiles” on page 176.

- **Install Section**

This section consists of the name of the installation program and its parameters. For installs of CID-enabled products, in addition to the name of the install program, it contains the parameters referring to the response file, product source and log file(s). It includes therefore additional support for CID-enabled install programs and their parameters that can be used easily. For non-CID installs only the name of the

executable or command and accompanying parameters may be included in this section as supported by the executable. The parameter section is not limited to the parameters used by CID but open for all parameters needed by any command.

- **FileSpecList Section**

Data files or product files to be sent to the client workstation are listed here. The files are replicated to the application site including the directory structure beyond the base target path. For CID installs, NetView DM/2 does not distribute any data files to the client workstation. Therefore, this section is omitted in the change file profiles for CID-enabled products. Instead, the CDM identifies an installation program and parameters/response files to be executed in the *Install Section*.

Using the dialog interface of NetView DM/2, you will be guided through the following panels:

- Catalog Change File panel
This is gathering the information of the catalog section.
- Installation panel
This is gathering the information of the install section. It includes the entries for the global statements.
- Files panel
This is gathering the information of the FileSpecList section.

If you are creating a change file for a CID-enabled product, all necessary files for the install, like the diskette image, the install program and the response file, have to be in the areas defined via the **SA:** or **SB:** parameters in the IBMNVDM2.INI file. These are the areas that are accessed by the client during an install using redirected drives.

NetView DM/2 offers a lot more install and change management options than those used by CID installs. You should review the product manuals to get an idea of how those options can be used. Most of these options depend on the fact that the CDM "knows" what was done at the client. When a CID install takes place, the CDM identifies that there is an install program to be executed at the client workstation. It invokes this executable and then waits for a return code to come back. The return code has to be an architected CID return code. The CDM does not know what is actually done at the client workstation. Therefore, you cannot execute installs in trial or service area or specify a CID install as removable. These functions can only be used if the FileSpecList section is used to specify which files are transferred to the client.

4.6.3 Create Change Files to Install CID-Enabled Products

There are two methods of creating change files. One way is to prepare an ASCII change file profile, the other way is to use the dialog interface to gather the profile information.

In this section, you will create the change files to install OS/2 Warp Connect as an example using change file profiles and LAN Server V5.0 as an example of using the dialog interface. First we will describe creating ASCII change file profiles. See 4.6.3.2, "Creating Change File Profiles with the Dialog Interface" on page 179 for creation of change files using the dialog interface.

Sample ASCII Profiles

Samples of ASCII change file profiles for the product installs described in this book are supplied in the NVDM2 directory on the sample code CDROM that comes with this book.

4.6.3.1 Change Files from Profiles

The profile CONNECT.PRO for the install of the OS/2 Warp Connect base code is created in this section. Following the description given here, you will be able to create other profiles. Remember to replace D: with the actual drive letter you are using.

1. Create a common directory for the change file profiles, for example **D:PROFILES**. See Figure 10 on page 45 for more details on the NetView DM/2 directory structure.
2. Use an ASCII editor and edit the contents of the sample profile shown below. You can either use the sample file provided or enter the lines.

```

TargetDir = C:OS2

Section Catalog
Begin
    ObjectType = SOFTWARE
    GlobalName = IBM.OS2.300.CONNECT.INST.REF.1
    Description = Installation-Procedure for OS/2 V3.00 WARP CONNECT
End

Section Install
Begin
    Program = SA:\Exe\connect\SEINST.EXE
    Pargs = /S:$(SourceDir) /B:C: /R:$(responsefile) /I:$(logfile1) /T:A:\
    SourceDir = SA:\IMG\connect
    ResponseFile = SA:\RSP\connect\$(WorkStatName).RSP
    LogFile1 = SB:\log\connect\$(WorkStatName).L1
End

```

Figure 32. ASCII Change File Profile for OS/2 Warp Connect

Each profile for CID-enabled products contains two major sections. The Catalog section specifies the object type, global name and a description of the change file to be created. Here, the global name is defined as "IBM.OS2.300.CONNECT.INST.REF.1" and the object type is software. The Install section describes the command to be executed and its parameters. The following variables are defined:

- **Program**

This variable is set to the name of the OS/2 Warp Connect install program (SEINST.EXE). The full path of the SEINST program is specified in order for the CC client to locate the program. SA: represents the shared A directory and SB: represents the shared B directory defined in the IBMNVDM2.INI on the CC server.

- **Pargs**

This variable defines the parameter list for the SEINST.EXE program. The parameter list can reference other variables defined in the profile such as ResponseFile, SourceDir, TargetDir. To reference the shared directories in the parameter list, you must use the variables SA: and SB: or another variable such as \$(SourceDir) whose assignment includes SA: or SB:.

- **ResponseFile**

This variable defines the path to the response file used for the SEINST program. It has to be stored in the shared area on the CC server. In its definition it uses the variable \$(WorkStatName).RSP to point to a client specific response file, because \$(WorkStatName) will be resolved with the CC client name. In opposition to the LCU

product definition, it is not possible to define that a default response file is to be used if a client specific response file cannot be found. You can define, however, one specific response file that is then used by all installs of this object.

- **SourceDir**

This variable points to the diskette image which resides in the shared A area on the CC server.

- **Logfile1**

This variable specifies the name of the log file that will be generated by the SEINST program. The log file will be written to the LOGCONNECT subdirectory in the CC server's shared B area to which the clients have write access.

- **TargetDir**

This variable is not part of the install section but of the global statements. It can, however, be used as a variable in the parameter section. It points to the drive and directory where the product will be installed.

Save this input under the name CONNECT.PRO.

3. Execute the CDM BUILD command.

The CDM BUILD command creates the change file and the catalog entry. It is invoked with the parameters <source file> and <target file>. If it is issued from the directory where the ASCII change file profiles reside, you do not have to specify a path for those. The target for the change file that is created is specified by either a fully qualified path or by the parameter **FS:**. The variable FS: is specified in the IBMNVDM2.INI of the CC server and it represents the File Services area. This directory is accessible for the CC client during an install. The change files should be placed there. As file name for the change file you can choose whatever you want though it makes more sense to use the same file name as the profile has with an extension specifying that this is a change file. Example for the CDM BUILD command:

```
D:
CD D:\PROFILES
CDM BUILD CONNECT.PRO FS:CONNECT.CHG
```

As an alternative to the last step where the NetView DM/2 line command is used to create the change file, you can use the dialog interface. Follow these steps to use the PM panels:

1. Start the NetView DM/2 dialog by entering **CDMD**. The **CDM catalog** window will be displayed. The CDM catalog displays all cataloged objects, that can be change files, flat files, etc. If this is the first time you activate the CDM dialog facility and you have not created any objects, you will see only one entry (**13IBM.49F4620.BASE.REF.2.ENGLISH**) which is created as an example during product install.
2. Select **FILE** from the action bar to display the menu.
3. Select **BUILD FROM PROFILE** from the menu to display the Build Change File screen.
4. Enter **D:PROFILESCONNECT.PRO** as the change file profile or use the **FIND** function.
5. Enter **FS:CONNECT.CHG** as the target file or use the **FIND** function that will automatically place you in the FS data area.
6. Select **BUILD** to build the change file.
7. You should receive the "ANXI5619 Build was successful but the change file contains only the install section" message. Select **OK**.
8. You should receive the **ANXI5670** message telling you that "IBM.OS2.300.CONNECT.INST.REF.1" was successfully built and cataloged".

If using NetView DM/2 V2.1, the messages will be slightly different, but the result is the same. The change file created by the BUILD function has a local name of CONNECT.CHG and is stored in the FS area. The catalog is updated accordingly.

4.6.3.2 Creating Change File Profiles with the Dialog Interface

This section describes how to use the dialog interface to create a change file, without having an ASCII file as input. A complete description of the entry fields used in this example can be found in the *IBM NetView Distribution Manager/2 Version 2.1 User's Guide*, SH19-5048-02. In this particular example we will define a change file profile for CID installation of LAN Server V5.0 Requester on the CC client. Your directory paths may be different, remember to reflect your own structure when following this scenario.

1. Start the NetView DM/2 dialog by entering **CDMD**.
2. Select **File** from the pull-down of the NetView DM/2 Catalog panel.
3. Select **Catalog > Change File > Refresh**.
4. The next panel is the **Catalog Change File (Refresh) panel** which contains the catalog information. There are three entry fields for the global name.

The change type has already been chosen with the pull-down selection.
Enter the following:

IBM.LS.500.REQ.INST

for the component name, and

1

for level. Adding the arbitrarily chosen level, the total global name will be:

IBM.LS.500.REQ.INST.REF.1

5. Select SOFTWARE as the object type.
6. Enter a file name for the change file. This file name is used by the CDM BUILD command as target file. Choose a file name that helps you to identify the change file. Use the FIND function or enter the full path for the file. In our example, we use LS50REQ.CHG as the change file name.
7. Click on the **Installation** button. The message "**File D:FSDATALS50REQ.CHG does not exist. Do you wish to continue?**" is displayed. Click on **YES**.
8. The installation panel appears where global statements and the Install section are to be entered.

Enter the following:

- Target directory:
C:IBMLAN
- Program:
SA:IMGLS50LANINSTR.EXE
- Parameters:
/REQ /R:\$(ResponseFile) /L1:\$(LogFile1) /L2:\$(LogFile2) /S:\$(SourceDir)
- Source directory:
SA:IMGLS50
- Response filename:
SA:RSPLS50\$(WorkStatName).RSP
- Log file:
SB:LOGLS50\$(WorkStatName).L1
- Press "down" arrow of the spin button to get another entry line for a second log file and enter:

SB:LOGLS50\$(WorkStatName).L2

End of phase is not needed.

9. Click on the **OK** button to return to the Catalog Change File panel. The Files panel is used to capture files for a cloning or replication installation. For a response file driven installation this panel remains empty. The **Export** and **Build** buttons are now enabled.
10. The **Export** button can now be used to create a valid ASCII change file profile.
11. Click on the **Build** button. You will receive the message that the Build was successful, but the Change File contains only the Install section.
12. Click on **OK** to return to the NetView DM/2 CDM-Catalog panel.

Now the change file is created and the catalog has a new entry. This object can now be sent to a client workstation to install LAN Server V5.0 requester function.

Chapter 5. Maintenance and Service

This chapter discusses the various ways of servicing installed products on client workstations. It will therefore describe the infrastructure for service and maintenance first and then give detailed information on how the different IBM products service is implemented.

5.1 Connecting a Client for Maintenance

In order to service a client workstation, the client has to be connected to the code server. As there is a cleanup performed at the end of the initial install (with IFSDEL and CASDELETE, please refer to 4.1, "CID Installation Commands" on page 79 for further information on these procedures) the client is no longer attached to the code server. To set up the connection to the code server once again, there are three possibilities:

1. Boot the client workstation with the two boot diskettes that were used for the initial install.
2. Boot the client from a separate maintenance partition that was prepared during initial install.

If you are using the NVDM/2 client function you will merely install the NVDM/2 client permanently on the client workstation. Therefore, there is no need to reattach the client to the CC server, as you already have a working connection after initial install.

If you are using LCU, you could also decide not to delete the client after the last installation is done and therefore have a permanent connection established to the client. We expect that nearly every installation provides the client with some kind of LAN attachment. This LAN attachment, for example with LAN server or NFS, can then be used to reattach the client to the code server while running an LCU command file as a network application.

5.1.1 Using Boot Diskettes

For a detailed description of how to create the boot diskettes please refer to 15.1.2, "SEDISK" on page 377 and the descriptions given in the server specific chapters concerning the LAN transport system that has to be added. Using the boot diskettes gives the advantage that by this boot from diskettes you can easily service the operating system itself, because there are no files in use or locked on the disk.

The disadvantage is that other service programs might need Presentation Manager available in order to run properly.

If originally individual client diskettes were distributed, they can now be used again. The administrator has to ensure that these diskettes are still available at the client workstation or they must be distributed again.

— Creating WARP Utility Diskettes —

Coming with OS/2 Warp Connect there is a utility called **BOOTDISK.EXE**. It can be found in the system setup folder. It is an easy way to create a set of boot diskettes. You need three 1.44M diskettes. The first two diskettes created are the "normal" boot diskettes, without any LAN transport system. The third diskette is a utility diskette. If you want to use these diskettes for remote install of service packs you need to add LAN transport to the second diskette as for diskettes created with SEDISK.

5.1.2 Maintenance Partition

This maintenance partition should have all files needed to connect to the code server. It will normally not have a Presentation Manager available. The operating system itself can be easily serviced. The use of a maintenance partition makes it necessary to have bootmanager installed. If you have any other LAN product running on the client that allows you to remotely execute a command on the client workstation, the reboot of the client from the maintenance instead of the production partition can be initiated by the remote administrator.

The decision whether a maintenance partition will be used has to be made before the initial install of a client; otherwise, a re-partitioning followed by a reinstall has to be done. The size of the maintenance partition has to be at least 7MB in order to run SEMAINT properly. For more information on SEMAINT, please refer to 4.1.1.5, "SEMAINT" on page 87. The maintenance partition can be used for other tasks, for example to back up essential files. Therefore, the size of the partition might be adapted. Other products might be useful on the maintenance partition, for example agent functions of LAN systems management products that are used in the LAN.

To install a maintenance partition for use with LAN CID Utility the following steps have to be executed:

1. Run SEMAINT with /T: parameter reflecting the drive letter of your maintenance partition.
2. Run THINLAPS.

3. Run THINIFS.
4. Optionally, install other products.

A detailed description of the command syntax refer to Chapter 4, "Client Installation Control Files" on page 79.

NVDM/2 Maintenance Partition

If you want to use a maintenance partition in an NVDM/2 environment, use the basic installation procedure NVDMCLT with the parameter /M (migration) to install the CC client in the maintenance environment. If the CC client is already installed in the production environment, the parameter /CO (configuration only) can be used.

5.1.2.1 BOOTOS2 Utility

You can setup a maintenance partition using a tool called **BOOTOS2**. This tool is available on the Developer Connection for OS/2 CD-ROM, and on the OS2TOOLS disk. This tool allows you to set up a maintenance partition from a running OS/2 system, with some useful enhancements compared to the one created with the OS/2 utility SEMAINT. You can choose between 'MINIMAL' which installs a fullscreen support, 'PM' to install a Presentation manager OS/2 Window or 'WPS' to integrate the availability of the workplace shell. A so installed maintenance partition including the workplace shell option requires about 9MB of harddisk space. We recommend to create a maintenance partition with a size of 20MB to be flexible for several service purposes. For the creation of this partition please refer to Chapter 8, "Auto-Partitioning the Hard Disk" on page 243. An ASCII documentation file comes with the product so we only explain the required parameters for a maintenance system with Presentation Manager support.

BOOTOS2 - Syntax

```
BOOTOS2 <SOURCE=drive:\path\>
        <TARGET=drive>
        <TYPE=PM|WPS>
        <NLS(Country,KBD,CodePage)>
        <2DISK[=drive] >
        <ABIOS>
        <REXX>
        <SWAP=drive:\path\>
        <TRACE[=drive:\path\file] >
        <HELP>
        <SYSED>
        <VDM>
        <FILE=[drive:\path\file] >
        <FORMAT[:FAT] >
        <FORMAT:HPFS>
        <FORMAT:NONE>
        <QUIET>
        <GA200|SP200|GA210|SP211|MR211|GA300>
```

- **SOURCE=drive:path**

This parameter indicates the location of the SYSINSTX program. If you omit this parameter BOOTOS2 will ask you for the installation diskettes. Replace drive:path with the redirected path the subdirectory DISK_0 of your OS/2 Warp Connect image.

- **TARGET=drive**

By default, BOOTOS2 will install the bootable system on a floppy disk in your A: drive. You can use the TARGET= argument to specify an alternate drive to install the bootable system on. This alternate drive can be another floppy or a partition on a harddisk. Any writable medium capable of being booted from can be a target.

Possible values for the parameter TARGET are single drive letters.

- **TYPE=PM|WPS**

BOOTOS2 will install a bootable system that will support PM applications if you select TYPE=PM. The bootable system will be accessed as a single OS/2 windowed command prompt. If you select TYPE=WPS the workplace shell will be available and some default folders are created.

- **REXX**

A base REXX-Support is installed.

- **SYSED**

The system editor is installed.

- **FORMAT:HPFS|FAT**

The selected target drive will be formatted with the given file system.

- **VGA**

The maintenance system is installed with standard VGA graphic resolution. We recommend to use this parameter to avoid graphic conflicts.

- **FILE=filename**

This option can be used to specify alternate files to be installed by BOOTOS2. The value of the option is the fully qualified file name of a text file; BOOTOS2 will examine each line in the file as follows:

- If the line is blank it will be ignored.
- If the line starts with a '*', it will be considered as a comment line and will be ignored.
- If the line starts with a '=', all text following the '=' will be considered the fully qualified file name of a file BOOTOS2 will copy to the OS2 directory of the target drive.
- All other lines will added unchanged to the CONFIG.SYS file of the target drive.

This text file can be compared to a response file.

— **Example for an Input Text File for BOOTOS2** —

```
=C:OS2DOS.SYS  
=C:\OS2\SETBOOT.EXE  
=C:\OS2\OS2ASPI.DMD  
DEVICE = \OS2\DOS.SYS  
BASEDEV=OS2ASPI.DMD
```

BOOTOS2 - Invocation Example

```
BOOTOS2 SOURCE=X:IMGCONNECTDISK_0
        TARGET=D:
        FORMAT:HPFS
        TYPE=PM
        SYSED
        REXX
        VGA
        FILE=X:\RSP\CONNECT\BOOTOS2.RSP
```

To install the LCU client on this partition you have to do the following steps:

1. Run THINLAPS
2. Run THINIFS

For a detailed description of the command syntax refer to Chapter 4, "Client Installation Control Files" on page 79.

The advantage of using a partition for maintenance purposes is the very quick booting process on this partition and then it can easily be used to access files on the productive system that are normally in use. For example if the file system on the productive system crashes you can run CHKDSK from the maintenance partition to recover it.

5.2 Introduction to Corrective Service Facility

This section describes how the Corrective Service Facility (CSF) is used for the distribution of OS/2 corrective service (called a Service Pak) for OS/2 using LCU from a server onto client workstations.

The purpose of CSF is to apply a Service Pak for the OS/2 operating system. This section shows how to use CSF to service OS/2.

Each product related to OS/2, for example the base operating system, LAN Server V5.0 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 OS2INSTALL 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 Service Pak for the OS/2 base operating system services the root directory, the OS/2 directory, and all subdirectories of the OS/2 directory.

The requirements of CSF to install OS/2 maintenance on an enterprise's workstations are similar to those required for the installation process. Service Pak diskette images reside on a server workstation and are available for client workstations to attach to and install service from. CSF uses a response file to determine maintenance installation characteristics, but this must not be confused with the response file used for the installation of OS/2 using a redirected drive.

Getting started with FSERVICE

The two most important Corrective Service Facility files for CID 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. They are delivered with the CSD. If you do not have them, you can get them from various FTP sites as well as from the OS2CSD tooldisk (package WKICKR).

- FSERVICE.EXE (used for remote unattended installation)
- RESPONSE.FIL (sample response file covering multiple scenarios)

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

In our lab, we used version F.127, which has been released 12-01-95, to install the FixPak 17 for OS/2 Warp Connect. You can easily identify the version by looking at FSERVICE. The output from DIR command should read:

```
FSERVICE.EXE  11-30-95  4.54a  269600 bytes
```

5.2.1.1 FSERVICE.EXE

CSF provides a program, FSERVICE.EXE, for the distribution of maintenance. As mentioned above this file is provided on the so-called kicker diskettes of the Service Pak. FSERVICE is an application similar to RSPINST in that it accepts input from a response file, and can read the Service Pak files from a redirected drive which removes the need to feed 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 Service Pak images. The images must have been prepared prior to service (see 5.3, "Servicing of OS/2 Products" on page 194). 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 contain all files and directories delivered with the CSD)

/T: Target directory

It specifies the directory from which the Service Pak 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 Service Pak 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 OS2INSTALLSERVICE.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

Note: Command line parameters override response file parameters.

5.2.1.2 The CSF Response File

The response file required by CSF 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 Service Pak diskettes, and also 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 to check all information coming with the Service Pak to find out if there is anything new.

There are several ways to create a valid Service Pak response file:

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

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

- **:SERVICE**
Indicates this to be a service. In other words the :SERVICE tag will install the necessary maintenance to the operating system.
- **:SYSLEVEL**
Indicates the syslevel files that should be serviced. If no parameter follows the SYSLEVEL 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 :SERVICE 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 Service Pak. That means that there is no BACKOUT possible.

- **:LOGFILE** < pathfilename >

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** < pathfilename >

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** < pathfilename >

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

5.2.1.3 Sample Service Pak Response File

The following response file will allow FSERVICE to perform a default Service Pak installation.

```
*Indicates this is a service
:SERVICE
*Indicates that all versions on all partitions will be serviced
:SYSLEVEL
*Indicates the archive path
:ARCHIVE C:\OS2\ARCH
*Indicates to update all files and exit
:FLAGS REPLACE_NEWER REPLACE_PROTECTED EXIT_WHEN_DONE
```

Figure 33. Sample Service Pak Response File

This default response file will service all products on the system. Exercise caution when using the default response file since it means all versions of OS/2 on all disks will be serviced (usually a problem if other OS/2 versions or OS/2 images are also installed on the system).

5.2.2 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 OS2INSTALL directory. If the file already exists then logging information will be appended.

5.2.3 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 Service Pak (as explained in 5.3.3.4, "Installation Method of FSERVICE.EXE" on page 199).

5.3 Servicing of OS/2 Products

The following section will describe the Service Paks and CSDs for the OS/2 products that were available when this book was written.

The descriptions cover only the steps for an LCU code server. For NetView DM/2, the change file profiles can be found on the sample code CDROM in the NetView DM/2 subdirectory, but they will not be described in detail, because the invocation for the install programs is the same.

The recommended CID directory structure as described in Chapter 2, "Recommended CID Directory Structure" on page 39 was extended to reflect the Service Paks and CSDs. The following figure gives an overview of the added directories.

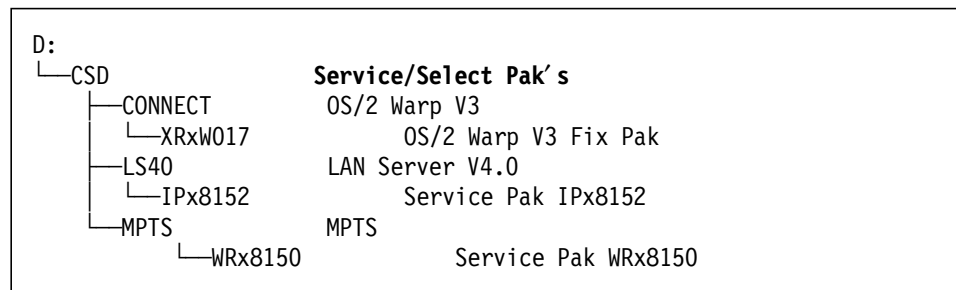


Figure 34. The Extended LCU Directory Structure for Service. The "x" is to be replaced with the character of the NLS version you are using.

If the Service Pak or CSD install creates a log file, the existing log directory of the base product is used to keep the directory structure smaller. The log

file has an extension different from the one used for the base installation so that they can be divided easily.

The CSDs and Service Paks names for OS/2 products may differ because of the different language versions of the base product. The third letter reflects the language of the base product, that is "0" for the US English versions, "G" for the German versions, "W" for the Swedish versions etc..

5.3.1 Service Pak and Select Pak

Service Paks and Select Paks are usually made for a special national language support (NLS) version of a software product. A CSD for another language should never be applied to your system, as this would give unpredictable results.

Rarely is a Service Pak/Select Pak declared NLS independent. The only example we can remember is the IP07005 for LAN Server, which only updated files common in all NLS versions of LAN Server.

5.3.1.1 Service Pak

A Service Pak is a cumulative CSD containing all updates since the base product. A later Service Pak always replaces earlier versions, so it is only necessary to apply the latest Service Pak available.

5.3.1.2 Select Pak

A Select Pak can contain updates for a part of a product and usually will have a requirement for an installed CSD level before the appliance of the Select Pak. Sometimes if the base level product is installed it can be necessary to first apply the latest available Service Pak, reboot and then apply the Select Pak(s).

It can also be the case that a 'manufacturing refresh' of a product is made, so if someone orders the product they get diskettes or CD-ROM with the Service Pak already applied. If this is the case the product can be at a high enough CSD level to apply a Select Pak directly.

An example of a 'manufacturing refresh' is if someone buys LAN Server V4.0 today (May 1996) they will in fact get LAN Server V4.01. For LAN Requester V4.0/LAN Server V4.0 the latest available CSD is the 8152 Service Pak, which can be applied to the LAN Server V4.0 as well as the LAN Server V4.01 version.

An example of applying a Select Pak directly is if someone has DB2/2 V1.01, which replaced DB2/2 V1.0, they can apply the DB2/2 V1.0 7022, 7023 and 7025 Select Pak's without applying Service Pak 7015 first.

5.3.2 Private Fixes

At times there are reasons to apply a so called private fix, when someone has a special problem and cannot wait for the next CSD for that software product.

The intention with these fixes is that only anyone who really needs it should install it and it is generally done manually. Usually these 'private fixes' come from IBM Support, but sometimes they are downloaded from an IBM BBS or CompuServe or Internet or any other such source.

Before applying such a fix care should be taken to save the old modules before they are replaced. The README or command file to install a fix normally helps the user with this.

The recommendation and sometimes a requirement is that at a later time, when a Service Pak/Select Pak is available, the private fix should be "backed out" and the original module(s) restored before the Service Pak/Select Pak is applied. If the fix module is dated later than the Service Pak/Select Pak (and it's not by mistake through the handling) it may be necessary to apply it again after the Service Pak/Select Pak is installed. If the fix is older than the Service Pak/Select Pak be happy to be updated to the latest version.

Most private fixes even if they are not CID enabled can be installed with the help of command files. If you do this remember that it is **not** officially supported. Please install it manually on a test machine first and ensure that it is running as expected on your software (and NLS version of the software in question). Then do a CID test installation and make sure at this stage to have routines in place to back out the fix(es). Test that it's really is working before you go ahead and update the clients in the production environment.

5.3.3 OS/2 Warp V3 Fix Pak

This section will explain the process to prepare the code server for OS/2 Warp V3 Fix Pak 17 distribution.

Please refer to the README files on the first Fix Pak diskette to check if there are any restrictions for the client workstations you want to service. This Fix Pak uses FSERVICE to apply itself.

It is not supported to service diskette images of OS/2 Warp V3 that are on the code server.

If the code server itself is serviced, care should be taken that only the code server's operating system is serviced and not the CID directory structure, as the OS/2 CID utilities are version dependent. Because of version specific utilities such as as XCOPY or FORMAT, the client boot diskettes used for installs of different versions have to be carefully separated.

5.3.3.1 Loading the Fix Pak Images

The Fix Pak for the US Version consists of 8 diskettes. Please check if there is an NLS version of the Fix Pak for your base version available. The Fix Pak diskettes can be loaded on the server using XCOPY. Load the diskettes by following these steps:

1. Create a suitable directory on the server using the MD command:

```
MD D:CIDCSDCONNECTXRxW017
```

You may want to replace the "x" with the character of your NLS version.

2. Copy all Fix Pak diskettes to the directory using XCOPY.

```
XCOPY A: D:CIDCSDCONNECTXRxW017 /S
```

The following tree is created:

```
D:CIDCSDOS2V21XRxW017
    \FIX
    \OS2
```

5.3.3.2 OS/2 Warp V3 Fix Pak Response File

A subdirectory for the response files has to be created:

```
MD D:CIDRSPCSDCONNECTXRxW017
```

Create a Fix Pak response file following the description in Chapter 5.2.1.2, "The CSF Response File" on page 191. The following figure shows what a working response file looks like:

```
:SERVICE
:SYSLEVEL
:ARCHIVE C:\OS2\FIXPAK
:FLAGS REPLACE_NEWER REPLACE_PROTECTED EXIT_WHEN_DONE
```

Figure 35. OS/2 Warp V3 Fix Pak Response File

If you are servicing your code server take care to only service the server's operating system and not accidentally the CID directory structure. Use the **SYSLEVEL** keyword as a pointer.

5.3.3.3 Creation of OS/2 Warp V3 Fix Pak LCU Command File

If both the OS/2 Warp V3 operating system and the OS/2 Warp V3 Fix Pak are to be installed on the client workstation using only one LCU REXX command file, the following changes have to be done to the default LCU command file provided on the sample code CDROM. The version of FSERVICE shipped with Fix Pak 17 is capable of handling the locked files, so there is no need to install a temporary maintenance partition or to boot from a maintenance partition to apply the Fix Pak. The install section has to be changed as shown in the following figure:

```

Do Forever
  Select
    when OVERALL_STATE = 0 then do
      if BootDriveIsDiskette() == YES then iterate      /* Check if booted from diskette*/
                                                         /* if it was, then goto state 1*/
      if RunInstall(x.semaint300) == BAD_RC then exit    /*Install maintenance system */
      if RunInstall(x.laps_prep) == BAD_RC then exit    /* Install LAPS prep system */
      if RunInstall(x.thinifs) == BAD_RC then exit      /* Install SRVIFS requester */
      if RunInstall(x.casinstl) == BAD_RC then exit     /* Install LCU */
      Call CheckBoot                                   /* Reboot if it was requested */
    end
    when OVERALL_STATE = 1 then do
      if RunInstall(x.seinst300) == BAD_RC then exit    /* Install operating system */
      if RunInstall(x.laps) == BAD_RC then exit        /* Install LAPS */
      if RunInstall(x.thinifs) == BAD_RC then exit     /* Install SRVIFS requester */
      if RunInstall(x.casinstl) == BAD_RC then exit    /* Install LCU */
      Call CheckBoot                                   /* Reboot if it was requested */
    end
    when OVERALL_STATE = 2 then do
      if RunInstall(x.xrxw017) == BAD_RC then exit     /* Install OS/2 Service Pak */
      Call CheckBoot                                   /* Reboot if it was requested */
    end
    when OVERALL_STATE = 3 then do
      if RunInstall(x.ifsdel) == BAD_RC then exit      /* Delete SRVIFS requester */
      if RunInstall(x.casdelet) == BAD_RC then exit    /* Delete LCU */
      Call Reboot                                      /* Reboot */
    end
  end
end
exit

```

Figure 36. LCU Command File Using SRVIFS for OS/2 Warp V3 and Fix Pak Install

The following figure shows the product definition part for FSERVICE.EXE in the LCU command file:


```

x.XRXW017 = 39                                /* structure index */
x.39.name=' OS/2 WARP 3 Service Pak'           /* product name */
x.39.statevar = 'CAS_' || x.39.name            /* state variable name */
x.39.instprog = 'x:\csd\CONNECT\xrxw017\fservice', /* fully qualified install program name */
               ' /s:x:\csd\CONNECT\xrxw017', /* - source directory */
               ' /T:C:\SERVICE', /* - service directory */
               ' /CID', /* - service via SEMAINT */
               ' /l1:l:\os2v21\' || client || '.162', /* - log file */
               ' /r:', /* - response file flag (auto selection) */
x.39.rspdir = 'x:\rsp\csd\os2v21\xrxw017'      /* response file directory */
x.39.default = 'default.rsp'

```

Figure 37. Product Definition Part for OS/2 Warp V3 Service Pak

5.3.3.4 Installation Method of FSERVICE.EXE

CSF will not automatically replace every file. The date, time and file name are checked to determine if the file on the Service Pak is different from the one installed. If the data matches, then no replacement of that file will occur. This eliminates the time involved in replacing all files on the system when only a subset have to be replaced.

At the end of this process, the SYSLEVEL files corresponding to the serviced products will be updated to reflect the new levels.

5.3.4 MPTS Upgrade

MPTS was shipped first with LAN Server V4.0 and has the SYSLEVEL WRx8000.

The currently available CSD for MPTS changes the SYSLEVEL to WRx8150.

The MPTS version which will be delivered with LAN Server V5.0 or WARP Server will have a higher syslevel.

5.3.4.1 Loading the MPTS CSD Image

As the diskettes are a CSD, they are placed in the D:CIDCSD directory. Example for the directory creation:

```
MD D:CIDCSDMPTSWRx8150
```

The image of the CSD diskettes can easily be loaded using the XCOPY command.

Example for the invocation:

```
XCOPY A: D:CIDMPTSWRx8150
```

5.3.4.2 MPTS CSD Response File

As the MPTS CSD WRx8150 uses the normal FSERVICE command to install, also the normal FSERVICE response file is used.

```
:SERVICE
:SYSLEVEL
:ARCHIVE C:\OS2\ARCH
:FLAGS REPLACE_NEWER REPLACE_PROTECTED EXIT_WHEN_DONE
```

Figure 38. Response File for MPTS CSD WRx8150

5.3.4.3 Creation of MPTS CSD WRx8150 LCU Command File

The install program used for the CSD WRx8150 of MPTS is the FSERVICE.EXE. There is no need to install a temporary maintenance system on the server, because the version of FSERVICE is capable of handling locked files. Therefore, the install can be invoked after the initial install without further preparation.

The following figure shows the product definition part for FSERVICE.EXE.

```
x.mpts_csd = 7 /* structure index */
x.7.name='MPTS CSD Installation on production system (PM)' /* product name */
x.7.statevar = 'CAS_' || x.7.name /* state variable name */
x.7.instprog = 'x:\csd\mpts\WRx8150\FSERVICE ', /* fully qualified install program name */
              ' /s:x:\mpts\mpts\WRx8150 ', /* - source directory */
              ' /ll:L:\laps\ || client ||'.log ', /* - log file */
              ' /r:' /* - response file flag (auto selection) */
x.7.rspdir = 'x:\rsp\csd\mpts' /* response file directory */
x.7.default = 'upgrade.rsp' /* default response file name */
```

Figure 39. Product Definition Part for MPTS CSD WRx8150

If an upgrade of a running version is to be executed, the client has to be reattached to the code server as described above. If only the upgrade is to be installed, no other product is necessary assuming that you have the client permanently attached to the server.

The following figure shows the install section of the LCU command file:

```

Do Forever
  Select
    when OVERALL_STATE = 0 then do
      if RunInstall(x.mpts_csd) == BAD_RC then exit          /* Install LAPS Upgrade */
    when OVERALL_STATE = 1 then do
      if RunInstall(x.ifsdel) == BAD_RC then exit           /* Delete SRVIFS requester */
      if RunInstall(x.casdelete) == BAD_RC then exit         /* Delete LCU */
      Call Reboot                                           /* Reboot */
    end
  end
end
exit

```

Figure 40. MPTS CSD WRx8150 Install Section

5.3.5 IBM Communications Manager/2 Version 1.1 Upgrade

For the upgrade from CM/2 V1.1 to V1.11 there is no separate Service Pak or CSD. A complete set of the product diskettes will be delivered.

An upgrade of a running CM/2 V1.1 client workstation is done with an install of CM/2 V1.11 specifying in the response file that this is a refresh.

5.3.5.1 Loading the CM/2 V1.11 Images

To load the images of CM/2 V1.11 the administrator has to create the proper subdirectories first, using the MD command. As the diskettes are the complete product, they should be placed under D:CIDIMG.

Example for the directory creation:

```
MD D:CIDIMGCM2V111
```

The images of CM/2 V1.11 can easily be loaded using the utility CMIMAGE provided by CM/2. CMIMAGE is described in 16.1.3, "Loading Communications Manager/2 Diskette Images" on page 384. CMIMAGE is invoked with the parameters <source> and <target>, and can be found on the first CM/2 diskette.

Example for the invocation:

```
CMIMAGE A: D:CIDIMGCM2V111
```

5.3.5.2 CM/2 V1.1 Response File for Upgrade

An example for an upgrade response file can be found on the first diskette of CM/2 V1.11. A working response file for the upgrade has to have at least the following:

```

CMUpdateType=2
CMInstallCurrentFeatures=2
CMStopCommunications=1

```

Figure 41. CM/2 V1.11 Upgrade Response File

5.3.5.3 Creation of CM/2 V1.11 LCU Command File

The install program used by CM/2 V1.11 is the same as for V1.1. Therefore, only the paths have to be changed.

The following figure shows the product definition part for CMSETUP. These definitions are included in the sample LCU command file that can be found on the sample code CDROM.

```

x.cm111 = 16                                /* structure index */
x.16.name='CM/2 1.11'                       /* product name */
x.16.statevar = 'CAS_' || x.16.name          /* state variable name */
x.16.instprog = 'x:\img\cm2111\cmsetup      /* fully qualified install program name */
              ' /cr                          /* - current response flag; if the
              ' ,                          /* response file is to be migrated, use
              ' ,                          /* /mg <path> <filename>
              ' ,                          /* /KL keylock password
              ' ,                          /* - installation log file CMRINST.LOG
              '/l1:l:\cm2111\' || client || '.11 /* - audit trail log file
              '/l2:l:\cm2111\' || client || '.12 /* - response file
              '/r '                          /* response file directory
x.16.rspdir = 'x:\rsp\cm2111'               /* default response file name */
x.16.default = 'mod3270.rsp'

```

Figure 42. Product Definition Part for CM/2 V1.11

If a client is to be installed for the first time CM/2 V1.11 is taken. If an upgrade of a running version is to be executed, the client has to be reattached to the code server as described above. CMSETUP is an install program that needs Presentation Manager to be present. If only the upgrade is to be installed, no other product is necessary assuming that you have the client permanently attached to the server. If you used the "seed" scenario to attach to the code server, you might use the cleanup utilities CASDELET and IFSDEL of the last queue.

The following figure shows the install section of the LCU command file:

```

Do Forever
  Select
    when OVERALL_STATE = 0 then do
      if RunInstall(x.cmsetup) == BAD_RC then exit          /* Install CM/2 Upgrade      */
    when OVERALL_STATE = 1 then do
      if RunInstall(x.ifsdel) == BAD_RC then exit          /* Delete SRVIFS requester   */
      if RunInstall(x.casdelete) == BAD_RC then exit        /* Delete LCU                 */
      Call Reboot                                           /* Reboot                     */
    end
  end
end
exit

```

Figure 43. CM/2 V1.11 Install Section for an Upgrade

5.3.6 LAN Server V4.0 Service Pak

For the LAN Server V4.0 product there is Service Pak available that changes the initial service level (IPx8000) to IPx8152. (The x stands for the specific NLS version of the CSD).

The CSD IPx8152 uses FSERVICE.EXE to get installed. There is no need to install a temporary maintenance system on the client. Therefore, the install of the CSD can follow right after the product install and a reboot done. The procedure described below is the same that was followed for the install of the MPTS Service Pak WRx8150. Please note the MPTS Service Pak WRx8150 is a prerequisite for the install of the LAN Server V4.0 CSD IPx8152.

Note that you always must use the FSERVICE delivered with a special CSD, whether it's for OS/2 or LAN Server or any other program. The README of the IPx8152 tells that you need to run a procedure called ARCHOFF before applying the CSD. This procedure can also be found on the CSD diskettes. It is invoked without any parameter, and it is therefore assumed that the administrator may include this product himself to the LCU command file.

5.3.6.1 Loading the Select and Service Pak Images

Create the proper directories on the code server using the MD command:

```
MD D:CIDCSDL40IPx8152
```

Copy all Service Pak diskettes to the directories using XCOPY:

```
XCOPY A: D:CIDCSDL40IPx8152 /S
```

5.3.6.2 LAN Server V4.0 Service Pak Response File

As IPx8152 is using FSERVICE.EXE it needs a response file to execute. A sample response file named RESPONSE.FIL can be found on the first Service Pak diskette. The following figure shows a working Service Pak response file.

```
:SERVICE
:SYSLEVEL
:FLAGS REPLACE_PROTECTED REPLACE_NEWER EXIT_WHEN_DONE
:ARCHIVE C:\OS2\LSARCH
```

Figure 44. LAN Server V4.0 Service Pak Response File

5.3.6.3 Creation of LAN Server V4.0 Service Pak LCU Command File

The following figure shows the product definition part for IPx8152.

```
x.ipx8152 = 40                                /* structure index */
x.40.name=' Lan Server 4.0 Service Pak 8152'   /* product name */
x.40.statevar = 'CAS_' || x.40.name            /* state variable name */
x.40.instprog = 'x:\csd\ls40\ipx8152\fservice.exe ', /* fully qualified install program name */
               ' /s:x:\csd\ls40\ipx8152        /* - source directory */
               ' /l1:x:\log\ls40\' || client || '.101 ', /* - log file */
               ' /r:'                          /* - response file flag (auto selection) */
x.40.rspdir  = 'x:\csd\ls40\ipx8152'          /* response file directory */
x.40.default = 'response.fil'                 /* default response file */
```

Figure 45. Product Definition Part for LAN Server V4.0 Service Pak IPx8152

To install the CSD during an initial installation, the install section of the LCU command file has to be changed as shown in the following figure:

```

Do Forever
  Select
    when OVERALL_STATE = 0 then do
      if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                                    /* if it was, then goto state 1*/
      if RunInstall(x.semaint300) == BAD_RC then exit /* Install maintenance system */
      if RunInstall(x.laps_prep) == BAD_RC then exit /* Install LAPS prep system */
      if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 1 then do
      if RunInstall(x.seinst300) == BAD_RC then exit /* Install operating system */
      if RunInstall(x.laps) == BAD_RC then exit /* Install LAPS */
      if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 2 then do
      if RunInstall(x.lansrva) == BAD_RC then exit /* Install LAN Server 4.0 */
      Call CheckBoot
    end
    when OVERALL_STATE = 3 then do
      if RunInstall(x.ipx8152) == BAD_RC then exit /* Install LS CSD IPx8152 */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 4 then do
      if RunInstall(x.ifsdel) == BAD_RC then exit /* Delete SRVIFS requester */
      if RunInstall(x.casdelet) == BAD_RC then exit /* Delete LCU */
      Call Reboot /* Reboot */
    end
  end
end
end
exit

```

Figure 46. LAN Server V4.0 Install Section for CSD IPx8152

5.3.7 IBM NetView Distribution Manager/2 Version 2.1 CSD

The latest CSD for NetView DM/2 V2.1 is CSD XR20466A dated July, 1995. It changes the SYSLEVEL to XR00002. The latest available Fix Pack is XREFP01, it changes the SYSLEVEL to XR00003.

An upgrade of a CC client workstation is done with a reinstall of NetView DM/2 V2.1 with the parameters /S: and /T: (and /L:) without specifying a response file. The install program, NVDMCLT.EXE, will detect an installed version on the client workstation and use the configuration it finds because there is nothing different specified in the product invocation.

Please check the README file 9 of the CSD for detailed information.

5.3.7.1 Loading the NetView DM/2 V2.1 CSD Diskettes

NetView DM/2 V2.1 CSDs support the update of the images that are already on the CC server. Therefore, the NetView DM/2 V2.0 diskette images can be updated while the CC server itself is upgraded to the new level when this is done with diskettes. If you choose to upgrade the images of the CC server via a change file, the CC server can then use these updated images to update itself.

5.3.7.2 NetView DM/2 V2.0 Change File for Upgrade

A change file for the update has to be created. Only the parameters for source, target and logging are needed.

A sample change file for a CC client upgrade named NDM2UPD.PRO can be found in the NVDM2 directory of the sample code CDROM.

If the images were updated on the CC server, all further installs will be executed with the newer version.

Chapter 6. Recovery Recommendations

This chapter discusses the implementation of backup and recovery for the CID environment.

The LAN CID Utility does not include any tools for backup or recovery tasks, even though, the installations performed via LCU touch vital system data and are therefore critical. Additionally, they may be initiated to occur overnight or over a weekend. Possible failures are:

- Incorrect product installation
- Invasion by a virus
- Installed fixes or product versions that do not run correctly
- Incompatibilities between the new product and the already installed ones

The failing condition can be detected by:

- Product installation error codes
- Operation of the system
- The user

The best way to prevent failure is extensive testing of the planned install. It should therefore be required to test the install on every type of workstation that you have, that means on all your hardware and on the existing configurations of already installed software.

Nevertheless, it is important to plan a way to restore a workstation to its previous working state. Most of the install programs of the CID-enabled products do not have a built-in capability of restoring a client workstation in case of a failure.

The following IBM products are capable of restoring the client workstation after a failed install:

- IBM Communications Manager/2 Version 1.1 and higher

If an error was encountered during the installation of CM/2, the installation continues up to the end and returns a "Reboot with callback" to the code server. When it is invoked again, it cleans up the client workstation by deleting all files that were already transferred.

Note: Due to this constellation it is recommended to install CM/2 V1.1 in an installation queue of its own.

- DATABASE 2 for OS/2

If specified in the response file with the keyword `DBBackupSystem DB2/2` backs up all files of an existing version of Database Manager, FFST/2 and User Profile Management. If the install fails, the saved files can be copied back to their origin. This is not done by the DB2/2 install itself but must be done manually or at least initiated separately if using procedures.

Note: The databases on the client are **not** backed up during the install of DB2/2. They must be saved during the backup of the user data. In case of an unsuccessful install, DB2/2 cleans up the client workstation by deleting all files that were already installed and deleting all objects that were already created.

Especially if more than one product is installed there might be no chance for an installation program to recover even if there are recovery routines. This is due to the fact that:

- Specific files changed by the installation of a CID-enabled product may not be easily identified because the installation process may indirectly change files through a system application programming interface (API) not related to the files.
- Inter-product dependencies may exist when two or more products update the same file. For example, assume that a product backs up CONFIG.SYS before altering it. If subsequent product installations also make changes to this file, this product cannot merely restore its backed up version because this action may invalidate CONFIG.SYS for those products that were installed after this product.

Backup/recovery procedures are therefore recommended. Their design can be different concerning their purpose:

For the client workstation:

- Saving the data for all products within a partition or drive at a code server. This type of recovery makes it unnecessary for each product to provide unique backup and recovery procedures and also removes all inter-product file dependencies.
- Saving the data only of the product that is to be replaced, and the related files of other components that are changed during the installation process, that is for example CONFIG.SYS. This type of recovery is only useful when replacing an already installed version. Though, this is the scenario that will hurt the end user the most. It may be difficult to determine which related files are changed during the installation process.

- Reinstall of the previous version. This leads to the loss of all customizations that were done on the client workstation if these were not captured.

Back Up Your System!

The backup/recovery procedures described here are only prepared to save the base operating system and related products. Thus, backup of the user data is not covered! Back up your users' data following your implemented backup procedures before you start an installation. Every CID installation initiated on a client workstation should be estimated as an extraordinary task that needs system backup comparable to a hardware feature change.

For the code server workstation:

- As the code server workstation is the critical system in the CID environment the best possible backup for this system should be used. The available backup/restore software is recommended. Your backup strategy for the LAN should be followed.
- To be able to reinstall all products of a client workstation the response files used for the initial install have to be kept.

The backup and restore procedures used in this chapter use the XCOPY utility which is capable of copying hidden files and directories, read-only files, and system files. If your backup/restore software supports a command line interface it can easily be integrated in the installation process by invoking it as the very first program of the install. Then invoke the backup/restore software to backup the files the way the RECOVER procedures get invoked in the following examples.

You may also want to use the built-in ARCHIVE utility of OS/2 Warp V3 to backup important system files with every system start. This function is activated via a push button in the desktop pull-down menu. It backs up to the subdirectory C:OS2ARCHIVE the files listed in the file OS2.KEY. The file OS2.KEY can be changed, and files can be added like PROTOCOL.INI, the CM/2 configuration files. OS2.KEY is read only, so if you want to change it, use the ATTRIB command first. Below the C:OS2ARCHIVE tree there will be three subdirectories for the last three copies of the files listed in the OS2.KEY file, numbered from 01 to 03. In the subdirectory 0X the configuration of WARP after the first reboot is saved. As you might want to change this to a backup of the files after the install, delete the 0X subdirectory and invoke the utility ARCINST. The 0X subdirectory is then recreated with the actual configuration. This could be done at the end of the first installation sequence

in order to prevent any recovery without connection to the server. It should be done at the end of the complete install, to have the final state of the workstation saved.

The ARCHIVE utility has the advantage that the user can access the saved files during system startup (while pressing ALT+F1 when the little block followed by OS/2 occurs in the upper left corner during system start, then a blue screen with several options including restore configuration files pops up) and can therefore easily recover the system after a failed installation. However, this feature is also a weakness for the CID install because it gives the user the possibility to avoid an installation and therefore brings the system to a state that is no longer controllable or, even worse, no longer accessible for CID installations.

When performing these tasks it is mandatory to define where the backup is located. The backup and restore files can be local to the client workstation or redirected to a server. When this is discussed, it should be considered that the amount of data produced by a backup may need remarkable DASD space on the client and/or on the code server. The capacity of the LAN transport system might also limit the backup options. Also the options of a seed or maintenance system have to be checked when discussing the backup procedure. Please refer to Chapter 5, "Maintenance and Service" on page 183 for a more detailed description.

Note

For detailed information on recovery while upgrading from OS/2 V1.3 to OS/2 V2.0 on a system running LAN Server V3.0 please refer to *IBM Network Transport Services/2 Redirected Installation and Configuration Guide*, S96F-8488, Appendix E, "System-Wide CID Recovery."

If MPTS is used the equivalent information is found in *LAN CID Utility Guide*, S10H-9742 in the chapter "System-Wide CID Recovery."

NDM/2 Options

If you are using NetView DM/2 as the software distribution manager, you can use the INSTALL options provided. These options include an install with backup to the NetView DM/2 CC server or an install into a temporary file/directory on the client workstation (trial or service area) that allows checking if the product is running on the system. These options can only be used for installation of non-CID-enabled products, as it is necessary for NDM/2 to know via the FileSpecList of the change file what was changed on the client. CID-enabled products use their own procedures to perform an install, and in this case NDM/2 is only used as a transport system.

If you are using NetView DM/MVS to control the installs performed by NetView DM/2 you can use the phase conditioning facility to automatically jump to a recovery procedure if an install is not successful. See the NetView DM/MVS manuals for details on the conditioning feature.

6.1 LCU Recovery Capability

The restore procedure for the failed client workstation needs to be attended at the remote installation site if the failure occurs during a critical function such as a REBOOT procedure. This event is termed a major failure. Note that REBOOT procedures are a normal part of some CID installation procedures, such as the LAN Server, CM/2, and OS/2 programs, to activate the newly installed code, such as the CONFIG.SYS and .INI statements.

The LCU recovery procedure described here can be automatically invoked by a bad return code received from an unsuccessful CID installation program, so it is not capable of coping with a major failure. See the following section for recovery after major failures.

You can customize the recovery procedure for each client workstation. A sample LCU REXX command file is specified in 6.1.1, "Sample LCU REXX Command File with Recovery" on page 212. The REXX file illustrates how this CID recovery can be automated without intervention at the remote site. The REXX file contains the statements for the case when the code server and the client workstation are the same. In 6.1.2, "REXX Recovery Sequence Preceding a Single CID-Enabled Product Installation" on page 213, the REXX file contains the statements for the case when the code server and the client workstation are not the same.

6.1.1 Sample LCU REXX Command File with Recovery

```
:vars
D:='Z:' /*variable substitution for backup drive*/
C:='C:' /*variable substitution for Maint. drive*/
:endvars
OVERALL_STATE = GetEnvironmentVars()
Do Forever
  Select
    when OVERALL_STATE = 0 then do
      'Xcopy C:\*.* D:\OLDR00T /H /T /R /O' /* save root files for OS/2 base*/
      if rc <> 0 then Exit
      Call CheckBoot
    end
    when OVERALL_STATE = 1 then do
      if BootDriveIsDiskette() == YES then iterate /*if boot install go to next state*/
      if RunInstall(x.semaint30) == BAD_RC then Call Recover1
      if RunInstall(x.mpts_maint) == BAD_RC then Call Recover1
      if RunInstall(x.thinifs) == BAD_RC then Call Recover1
      if RunInstall(x.casinst1) == BAD_RC then Call Recover1
      Call CheckBoot
    end
    when OVERALL_STATE = 2 then do
      'Xcopy C:\*.* D:\BACKUP /S /E /H /T /R /O' /*save all partition fls for OS/2 base*/
      if rc <> 0 then Exit
      Call CheckBoot
    end
    when OVERALL_STATE = 3 then do
      if RunInstall(x.seinst30) == BAD_RC then Call Recover2
      if RunInstall(x.mpts_prod) == BAD_RC then Call Recover2
      if RunInstall(x.thinifs) == BAD_RC then Call Recover2
      if RunInstall(x.casinst1) == BAD_RC then Call Recover2
      Call CheckBoot
    end
    when OVERALL_STATE = 4 then do
      if RunInstall(x.cm111) == BAD_RC then Recover3
      Call CheckBoot
    end
    when OVERALL_STATE = 5 then do
      if RunInstall(x.lanreqa) == BAD_RC then Recover3
      Call CheckBoot
    end
    when OVERALL_STATE = 6 then do
      if RunInstall(x.casdelet) == BAD_RC then Exit
      if RunInstall(x.ifsdel) == BAD_RC then Exit
      Call Reboot
    end
  end
end
end
```

Figure 47 (Part 1 of 2). LCU REXX Command File with Recovery

```

Recover1:
'ERASE C:\*. * <X.FIL'                                /* to erase root files before restore*/
'XCOPY D:\OLDROOT C:\ /H /R /T /O '                    /* restore root files */
'ERASE D:\OLDROOT\*. * <X.FIL'                          /* cleanup saved root files*/
if RunInstall(x.casdelet) == BAD_RC then Exit
if RunInstall(x.ifsdel) == BAD_RC then Exit
EXIT
Recover2:
'ERASE C:\*. * <X.FIL'                                /* to erase root files before restore*/
'XCOPY D:\BACKUP C:\ /S /E /H /T /R /O'                /* restore all files */
'XCOPY D:\OLDROOT C:\ /H /R /T /O '                    /* restore root files */
'ERASE D:\OLDROOT\*. * <X.FIL'                          /* cleanup saved root files*/
if RunInstall(x.casdelet) == BAD_RC then Exit
if RunInstall(x.ifsdel) == BAD_RC then Exit
CALL Reboot                                            /* reboot restored system */
Recover3:
if RunInstall(x.semaint30) == BAD_RC then Exit          /* build maintenance system*/
if RunInstall(x.mpts_maint) == BAD_RC then Exit
if RunInstall(x.thinifs) == BAD_RC then Exit
if RunInstall(x.casinstl) == BAD_RC then Exit
CALL REBOOTANDGOTOSTATE (7)                            /* Reboot Maintenance System*/
when OVERALL_STATE = 7 then do
'ERASE C:\*. * <X.FIL'                                /* to erase root files before restore*/
'XCOPY D:\BACKUP C:\ /S /E /H /T /R /O'                /* restore all files */
'XCOPY D:\OLDROOT C:\ /H /R /T /O '                    /* restore root files */
'ERASE D:\OLDROOT\*. * <X.FIL'                          /* cleanup restored root files*/
'C:\Service\Lapsdel'                                    /* Run LAPS cleanup of itself*/
'ERASE C:\Service\*. * <X.FIL'                          /* delete files in service directory */
'RD C:\Service'                                         /* remove service directory*/
if RunInstall(x.casdelet) == BAD_RC then Exit
if RunInstall(x.ifsdel) == BAD_RC then Exit
CALL Reboot                                            /* reboot restored system */
end

```

Figure 47 (Part 2 of 2). LCU REXX Command File with Recovery

6.1.2 REXX Recovery Sequence Preceding a Single CID-Enabled Product Installation

When installing CID-enabled products separately, you can use the following series of statements (using a bootable OS/2 maintenance system) to back up the system before the actual installation of the CID-enabled product occurs. These statements assume that there are no ACL files on the current system.

```

:vars
D:='Z:' /*variable substitution for backup drive*/
C:='C:' /*variable substitution for Maint. drive*/
:endvars
OVERALL_STATE = GetEnvironmentVars()
Do Forever
  Select
    when OVERALL_STATE = 0 then do
      'XCOPY C:\*. * D:\OLDROOT /H /T /R /O' /* save root files for OS/2 base*/
      if rc <> 0 then Exit
      Call CheckBoot
    end
    when OVERALL_STATE = 1 then do
      if RunInstall(x.semaint30) == BAD_RC then Call Recover1
      if RunInstall(x.mpts_maint) == BAD_RC then Call Recover1
      if RunInstall(x.thinifs) == BAD_RC then Call Recover1
      if RunInstall(x.casinst1) == BAD_RC then Call Recover1
      Call REBOOTANDGOTOSTATE(2)
    end
    when OVERALL_STATE = 2 then do
      'XCOPY C:\*. * D:\BACKUP /S /E /H /T /R /O' /* save all partition files for OS/2 base*/
      if rc <> 0 then Exit
      'ERASE C:\*. * <X.FIL' /*cleanup root*/
      'XCOPY D:\OLDROOT c:\ ' /* restore root files for reboot*/
      if rc <> 0 then Exit
      Call REBOOTANDGOTOSTATE(3)
    end
    when OVERALL_STATE = 3 then do
      'C:\Service\Lapsdel' /* run LAPS cleanup of itself */
      'ERASE C:\Service\*. * <X.FIL' /* delete files in service directory */
      'RD C:\Service' /* remove service directory*/
    end
  end
end
.
.
exit

```

Figure 48. Sample Recovery REXX Statements

6.1.3 LCU Recovery for Major Failures

Failures that occur during a REBOOT function in an installation run by a CID installation sequence (such as an LCU command file) are considered major failures. For example, this can be a power failure or a LAN problem. Since the system is inoperable, someone must be present to attend the CID recovery, assuming that hardware related problems are solved and the client workstation is able to boot and to connect to the LAN. This event requires the following steps:

1. At the code server, create a command file on the code server that is to be accessed by a client workstation in a major recovery state. Assign any name for this command file. The following example uses the file name RECOVER. The command file contains the following REXX statements:

```
:vars
D:= ' z:' /*variable substitution for backup drive*/
C:= ' c:' /*variable substitution for backup drive*/
:endvars
'ERASE C:\*.* <X.FIL' /* erase root files before restore*/
'XCOPY D:\BACKUP C:\ /S /E /H /T /R /O' /* restore root files */
'XCOPY D:\OLDROOT C:\ /H /T /R /O' /* restore root files */
CALL Reboot /* reboot restored system */
```

Since the reboot failure in the installation sequence may have occurred before data was saved in D:BACKUP, any error returned from the statement

```
XCOPY D:BACKUP C: /S /E /H /T /R /O
```

assumes that the installation sequence saved data only from D:OLDROOT. This save should not affect the restoration of the workstation to its prior operating environment since the original C: files remain intact.

2. At the client, the user inserts the boot diskettes created for LCU. The name of the client workstation must match the name (RECOVER) of the recovery command file on the code server. The command file runs and the system is restored. The only user action at the failed workstation is to insert the boot diskettes. No knowledge about the failed system or recovery procedures is required.

If this manual CID recovery fails, then the system must be built as in a first-time installation.

Chapter 7. Remote Multiple Printer Support

This chapter 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.

7.1 Introduction

The remote multiple printer installation program (RINSTPRN) for OS/2 was written during residencies at the Boca Raton ITSO. Its main purpose is installation of the printer related issues at the time of initial OS/2 installation. It will run on OS/2 V2.1, OS/2 V2.11 and OS/2 Warp V3.

The application makes it possible to install multiple printers and queues via a response file instead of going through many dialogs. 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 etc. during the automated process. It also allows the definition of network queues, and the definition of WIN-OS/2 printers.

The application executes either from the UserExit or ExtendedInstall statements of the OS/2 response file or locally from the command line. Another way to use this application is to define a RMPI program entry to the LCU. See 7.7, "Using LCU" on page 241.

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.

The drivers that need to be installed can be read from either drive A: or B:. The program then prompts for diskette insertion on that specific drive. For any other drive letter the program looks in the specified directory for subdirectories with the names PMDD_1 to PMDD_n, which should contain images of the original printer driver diskettes.

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

Note: Already installed printer drivers will automatically be replaced by the program.

The sample at the end of this chapter describes a rather complex configuration of queues and printers and provides an overview showing the strengths of this program.

7.2 Definitions for Remote Printer Installation

This section describes the objects you can specify for remote installation, using the *RINSTPRN* program.

7.2.1 Definition of Local Printers and Queues

Local printers and queues are defined using the keywords *Printer*, *PrinterName*, *PrinterDesc*, *PrinterPort*, *Queue*, *QueueName*, *QueueDesc* and *QueueProc*. A detailed description of these keywords is available in 7.5.1, “Keywords Description” on page 227. All these keywords have a numeric suffix. The range for the printer keywords is between 1 and 20, and for the queue keywords it is between 1 and 60. This sets limits for the number of printers and queues which may be installed. A printer is defined by specifying the number of printer drivers the printer is connected to. Using the *PrinterPort* keyword, the port it prints to can be set. Besides these definitions, a printer may have a name and a description. If none is specified, they are generated, following these rules:

Printer Name: Name of the first printer driver plus the number of the port. For example, a printer connected to LPT2: using “PSCRIPT.DRV”, becomes “PSCRIPT2”.

Printer Description: The descriptive name of the printer driver concatenated with the word “at” and the connected port, is used. For example, a printer using the driver “IBM Personal Page Printer II-31” and connected to LPT2:, the description is set to “IBM Personal Page Printer II-31 at LPT2”.

Queues are created by specifying the number of the printer they connect to. Additionally the number of the printer driver in the *Printer* statement may be given. If no printer driver is specified, the first defined for the printer is used. A queue may have assignments to more than one printer. Besides these values, a queue may also have a name, a description and a queue processor. If none is specified, they are generated following these rules:

Queue Name: Name of the driver plus the number of the port the assigned printer is connected to. For example, for a queue connected to a printer on LPT2: using "PSCRIPT.DRV", the name is set to "PSCRIPT2".

Queue Description: The descriptive name of the driver of the assigned printer, concatenated with the word "at" and the port of the assigned printer, is used. For example, for a queue connected to a printer on LPT2: using the driver "IBM Personal Page Printer II-31," the description is set to "IBM Personal Page Printer II-31 at LPT2".

Queue Processor: If the driver of the assigned printer is "PLOTTERS.DRV", "PMPLOT" is used; in all other cases "PMPRINT" is used.

7.2.2 Definition of Printer and Job Properties

Special printer and job properties for a printer and a queue are specified using the *Properties* keyword in addition to the printer and the queue number, a property file name is specified here. A property file contains printer and job property definitions. It is created using the **BACKPRN** program. This program is described in 7.4.1, "Backup of Printer and Job Properties" on page 224. Because printer and job properties are related to each other, it is only possible to restore both of them together.

7.2.3 Definition of Network Queues

Disclaimer

When network queues are created, no network printer icon but a printer icon is shown on the OS/2 Workplace Shell. Even if the printer behaves like a network printer object, the contents of the queue on the network are not shown.

Network queues are created using the *NetQueue* keyword. The name of the remote computer, the remote queue, the network type and the printer driver have to be specified as parameters. A local queue as well as a local printer is created, which is linked to the queue on the remote computer, without using any local port. The name of the local queue and printer is the same as the remote queue name. The description, the queue processor and a property file may be specified as additional parameters.

7.2.4 Definition of WIN-OS/2 Printers

In addition to OS/2 printers, queues and drivers, WIN-OS/2 drivers can be installed also. This is done using the *WinPrinter* keyword. As parameters, either a list of OS/2 printer numbers, or the WIN-OS/2 driver name and the port is passed. If an OS/2 printer is specified, the WIN-OS/2 printer driver corresponding to the OS/2 printer driver is installed and the port of the OS/2 printer is used. For example, if the OS/2 printer uses port LPT2:, the WIN-OS/2 printer uses port LPT2.OS2.

7.2.5 Miscellaneous Definitions

In addition to the things mentioned above, other definitions can be made too. This includes additional printer drivers, using the *AdditionalPrinter* keyword, as well as communication port settings, using the *CommPort* keyword, and default definitions, using the *DefaultPrinter* and *DefaultQueue* keywords.

7.3 Remote Printer Installation Program

The program to be called has the name **RINSTPRN**. The following keywords can be used on the command line:

```
/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 ":" 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 of the 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 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.

Important

This keyword is OS/2 version dependent.
Specify CONTROL.INF for installation on OS/2 V2.1 and OS/2 Warp V3.

The default value for this parameter is CONTROL.INF. This file resides in the OS2MDOSWINOS2SYSTEM 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 OS2MDOSWINOS2SYSTEM 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 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. The following example is valid for installation on OS/2 V2.1 and OS/2 Warp V3 since we specify the **CONTROL.INF** file for **/WPR** keyword.

```
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
```

7.4 Backup and Restore of 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.

7.4.1 Backup of 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 2 types of properties are closely related to each other, so that 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 by either using the **RESTPRN** program (see 7.4.2, “Restoring Printer and Job Properties” on page 225), or the remote printer installation program **RINSTPRN** (see 7.3, “Remote Printer Installation Program” on page 220).

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

<printer-name>      is the name of the printer to copy the printer properties from
<queue-name>      (optional) is 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>         is 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.

Note: In order to create a property file, the printer driver installed has to be either an OS/2 2.x or OS/2 1.3 CSD 5050 driver.

7.4.2 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 <file-name> [<printer-name>[.<queue-name>]]

<file-name>          is the name of the property file
<printer-name>  (optional) is 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) is 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
```

Note: In order to restore a property file, the printer driver installed has to be either an OS/2 2.x or OS/2 1.3 CSD 5050 driver. The target machine must be at the same level of OS/2 and NLS support as the backup machine.

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

<queue-name>	is the name of the queue for that the status has to be changed or displayed
/H[OLD]	to hold the queue
/R[LEASE]	to release the queue

For Example: CHGQUE PSCRIPT1 /H

7.5 Printer Response File Keywords

The following keywords are available for the remote installation of printers. With these keywords up to 20 printers and 60 queues, and an unlimited number of network printers can be defined for one workstation. Most of the keywords are suffixed. The keywords are listed in alphabetical order. Each of the following keywords is explained in detail on the following pages.

AdditionalPrinter

CommPort

DefaultPrinter

DefaultQueue

NetQueue

Printer

PrinterDesc

PrinterName

PrinterPort

Properties

Queue

QueueDesc

QueueName

QueueProc

WinPrinter

Each keyword is optional. Keywords can be used in any order. Only one keyword is allowed per line. Keywords are followed immediately by an "=" (equal) sign and a keyvalue. Blanks are not allowed between keyword, equal sign and keyvalue. There is no concatenation of lines. Comment lines start with an "*" (asterisk). Suffixable keywords *must* have a valid suffix.

Note:

The **CommPort** keyword must be suffixed with a value between 1 and 3. The CommPort suffix corresponds to the COM address. So CommPort1 corresponds to COM1.

The **Printer** keywords (Printer, PrinterPort, PrinterName and PrinterDesc) must be suffixed with a value between 1 and 20. So, Printer5 corresponds to PrinterPort5 and PrinterName5, etc.

The **Queue** keywords (Queue, QueueName, QueueDesc, QueueProc) must be suffixed with a value between 1 and 60. So Queue12 corresponds to QueueDesc12, etc.

7.5.1 Keywords Description

AdditionalPrinter: Define additional printer drivers.

Specifies which printer drivers should be installed in addition to the ones defined on the *Printer* keyword(s) within this same response file. This can provide an easy scenario for updating printer drivers over those already installed, as well as defining additional printer drivers, which may be used later on a new installation. The keyword is followed by an equal sign and the keyvalue for the entry in the printer description file (PRDESC.LST). The keyvalue is the line number of the specific line in PRDESC.LST.

More than one keyvalue can be defined on the AdditionalPrinter statement.

The natural limit on the number of keyvalues is the line length of 200 characters. If more than one keyvalue has been defined, values must be separated by either one or more blanks and / or a comma.

Note: A description of the PRDESC.LST is shown in Appendix C, "OS/2 Response File Keywords," section C.3, "Printer Description Table for AdditionalPrinters and DefaultPrinter Keywords."

```
Additionalprinter=<driver-number> [<drivernumber> [...]]
```

<drivernumber> is the number of the printer driver in PRDESC.LST

For example: AdditionalPrinter=24 27

CommPort: Specify communication port settings.

The parameters are positional and separated by a comma. If a parameter is omitted the positional comma still must be placed (see example):

```
CommPort<x>=<b>,<p>,<d>,<s>,<h>
```

<x> Communications port number between 1 and 3

 Baud rate

Valid baud rates are: 110, 150, 300, 600, 1200, 1800,
2400, 3600, 4800, 7200, 9600, 19200

<p> Parity, O = Odd, E = Even, N = None

<d> Data bits, valid values are 5 to 8

<s> Stop bits, valid values are 1, 1.5 and 2

<h> Handshake, H = Hardware, N = None

Default is 1200,0,7,1,N

For example: CommPort1=9600,N,8,1,H

CommPort2=9600,,, results in (9600,0,7,1,N)

CommPort3=4800,,8,,H results in (4800,0,8,1,H)

DefaultPrinter: Define a default printer.

Specifies the suffix number from the *printer* keyword of the printer that should be the default printer.

```
DefaultPrinter=<printer-number>
```

<printer-number> valid printer suffix number

For example: DefaultPrinter=1

DefaultQueue: Define a default queue.

Specifies the suffix number of the *queue* keyword of the queue that should be the default queue.

```
DefaultQueue=<queue-number>
```

<queue-number> valid queue suffix number

For example: DefaultQueue=1

NetQueue: Define a link to a queue on a network computer.

This keyword allows the creation of a link to a network queue, without using any local port. Every use of the keyword defines a local printer and a local queue, which is linked to a queue on a remote computer. The network type, server name, queue name and printer driver have to be specified as parameters to this keyword. Optional arguments are the printer description, the queue processor and a printer and job properties file. The usage is:

```
NetQueue=<network-type>:<computer-name><queue-name>, <printer-driver>
        [,<printer-description>"] [,<queue-processor>] [,<property-file>]
```

<network-type> is the LAN type you have installed, possible values are:
 LS for IBM LAN Server and NW for Novell NetWare

<computer-name> the name of the computer, the queue resides on

<queue-name> the name of the queue on the remote computer

<printer-driver> the printer driver number (see Printer keyword)

optional arguments are:

<printer-description> description for the printer, enclosed in double quotes

<queue-processor> name of the queue processor (see QueueProc keyword)

<property-file> name of the property file (see Properties keyword)

For example:

Netqueue=LS:\\PRNTRSRV\\PSCRIPT1, 124, "IBM 4029 Laser Printer", IBM4029.PJP
creates a local printer and queue named PSCRIPT1, with the printer driver
"IBM 4029 LaserPrinter 10L: IBM 4029 LaserPrinter 10L (IBM4019.DRV)",
which is linked to the queue PSCRIPT1 on the computer PRNTRSRV using
IBM LAN Server, and use the printer and job properties specified
in IBM4029.PJP.

Note that a local printer and a local queue, with the name specified as the remote queue name, are created. Therefore the name used *must not* be used on any other local printer or local queue. Also the LAN administrator must provide the appropriate port assignment in the logon details for the target client.

Printer: Define a printer.

With the keywords *Printer1* to *Printer20* up to 20 printers can be defined. The keyword is followed by an equal sign and the keyvalue for the entry in the printer description file (PRDESC.LST). The keyvalue is the line number of the specific line in PRDESC.LST.

More than one keyvalue can be defined per printer statement. This makes it possible to define different drivers per printer.

The natural limit on the number of keyvalues per printer is a line length of 200 characters. If more than one keyvalue has been defined, values must be separated by either one or more blanks and/or a comma.

Note: A description of the PRDESC.LST is shown in Appendix C, “OS/2 Response File Keywords,” section C.3, “Printer Description Table for AdditionalPrinters and DefaultPrinter Keywords.”

```
Printer<x>=<driver-number> [<driver-number> [...]]

<x>                printer number between 1 and 20
<drivernumber>     is the number of the printer driver in PRDESC.LST

For example:  Printer1=24 27
              Printer2=87
```

PrinterDesc: Define a printer description.

In case this statement has been omitted for a specified printer the program takes the description found in PRDESC.LST and adds the PrinterPort to it, for example “IBM 4202 Proprinter III XL at LPT2”.

```
PrinterDesc<x>=<printer-description>

<x>                printer number between 1 and 20
<printer-description> any ASCII string up to 60 character

For example:  PrinterDesc1=My favorite printer IBM 4019
```

PrinterName: Define a printer name.

If the name is longer than eight characters it will be truncated. In case this statement has been omitted for a specified printer the program takes the name of the first driver used by this printer and adds the number of its PrinterPort, for example “IBMNULL1”, “PSCRIPT2”.


```
PrinterName<x>=<printer-name>
```

```
<x>          printer number between 1 and 20  
<printer-name> any ASCII string up to 8 character
```

```
For example: PrinterName1=MY4019
```

Note: If two printers use the same driver, one on LPT1 and the other one on COM1, this yields a duplicate name. As a result the second printer will *not* install.

PrinterPort: Define a printer port.

```
PrinterPort<x>=<port>
```

```
<x>          printer number between 1 and 20  
<port>       printerport = LPT1 - LPT12, COM1 - COM3
```

```
For example: PrinterPort1=LPT1  
PrinterPort2=COM1
```

Properties: Define printer and job properties.

Specifies a property file for a printer and a queue, connected to the printer. The property file is created using the **BACKPRN** program, described in 7.4.1, “Backup of Printer and Job Properties” on page 224. The printer driver, which is used when creating the property file, as well as the driver used by the printer and queue have to match. For querying the printer driver stored in the file and the one for the printer and queue, use the **RESTPRN** program, described in 7.4.2, “Restoring Printer and Job Properties” on page 225. The usage is:

```
Properties=<printer-number>, <queue-number>, <property-file>
```

```
<printer-number> is the number of the printer as defined with the Printer keyword  
<queue-number>  is the number of the queue as defined with the <Queue> keyword  
<property-file> is the name of the property file, created with the BACKPRN program
```

```
For Example: Properties=1, 2, PSCRIPT.PJP  
creates printer and job properties defined in the PSCRIPT.PJP file for  
Printer1 and Queue2
```

Queue: Define a queue.

Describes the connection of a queue to one or more printers. The first parameter is the number of the printer, the second optional parameter defines the driver index within the printer definition which should be used by this queue. This keyword may occur more than once for a queue.

```
Queue<y>=<printer-number> <driver-index>

<y>                queue number between 1 and 60
<printer-number>   the suffixed number of the appropriate printer
<driver-index>     index in the driverlist of the Printer statement

For example: Queue1=1,2
              Queue1=2
```

This example connects Printer1/Driver2 and Printer2 to Queue1.

QueueDesc: Define a queue description.

Specifies the description for a queue. In case this statement has been omitted for a specified queue the program takes the description found in PRDESC.LST and adds the PrinterPort to it, for example, "IBM 4202 Proprinter III XL at LPT2". If no printer is connected to the queue, the description will be blank.

```
QueueDesc<y>=<queue-description>

<y>                queue number between 1 and 60
<queue-description> ASCII string up to 60 character

For example: Queue1=Queue for my favorite printer IBM 4019
```

QueueName: Define a queue name.

In case this statement has been omitted for a specified queue the program takes the name of the driver used by this queue and adds the number of its PrinterPort, for example, "IBMNULL1","PSCRIPT2".

Note: If a queue is not connected to a printer or its name has already been generated for another queue the queue will *not* install.

QueueName<y>=<queue-name>

<y> queue number between 1 and 60
<queue-name> ASCII string up to 8 character

For example: QueueName1=MY4019Q

QueueProc: Define a queue processor.

If this keyword is not defined for a queue, either *PMPRINT* or *PMPLLOT* will be used as queue processor, depending on the printer driver used.

QueueProc<y>=<queue-processor>

<y> queue number between 1 and 60
<queue-processor> queue processor name (PMPRINT or PMPLLOT)

For example: QueueProc1=PMPRINT

WinPrinter: Define a WIN-OS/2 printer.

Creates a WIN-OS/2 printer and installs the appropriate WIN-OS/2 driver. There are two different parameter specification types for this keyword. Either a list of OS/2 printers or a WIN-OS/2 printer driver and port may be specified. One occurrence of this keyword cannot have both usage types, but the keyword may be used multiple times.

If an OS/2 printer list is given, the first printer driver specified for the printer will be used. The program looks in the file *DRVMAP.INF* for the appropriate WIN-OS/2 printer driver. The port used is determined from the port used by the OS/2 printer as well. If an *LPT<x>:* port is used there, the WIN-OS/2 port is *LPT<x>.OS2*. Using these ports enables the OS/2 spooler to spool the WIN-OS/2 jobs. If a *COM<y>:* port is used, no OS/2 spooling occurs, and the WIN-OS/2 port is *COM<y>:* too.

Note: If WIN-OS/2 printer drivers are to be installed, the files *DRVMAP.INF* and *CONTROL.INF* for OS/2 V2.1x or OS/3 should be copied from the directory *OS/2MDOSWINOS2SYSTEM* on the WIN-OS/2 drive from a machine with the same SYSLEVEL to a shared drive on the installation server.

If a WIN-OS/2 printer driver is specified as a parameter to the keyword, a WIN-OS/2 port have to be specified as well. Legal ports are *LPT<x>:*,

LPT<x>.OS2 and *COM*<y>, where <x> has to be a number from 1 through 12 and <y> has to be a number from 1 through 3. Note that the WIN-OS/2 printer driver name has to be enclosed in double quotes. A "*" character may be used as a wildcard at the end of the driver name. A list of valid WIN-OS/2 printer driver names can be found in the file *CONTROL.INF* for OS/2 V2.1x. and OS/2 V3.0.

```
WinPrinter=<printer-number> [, <printer-number>]
or
WinPrinter="<printer-driver>", <port>
```

<printer-number>	is the number of an OS/2 printer defined prior
<printer-driver>	is a WIN-OS/2 printer driver name, which may have a trailing * as wildcard character
<port>	is a WIN-OS/2 printer port

For Example:

```
WinPrinter=1, 2
creates 2 WIN-OS/2 printers, with driver and port according to Printer1
and Printer2
```

```
WinPrinter= "IBM Personal Page Printer II-031*", LPT4.OS2
creates a WIN-OS/2 printer named [Postscript Printer]
using the driver PSCRIPT.DRV on port LPT4.OS2
```

7.6 Sample Printer Response File

The following scenario does not reflect a regular installation, but shows some capabilities of the OS/2 remote multiple printer installation.

Three local printers are connected to a workstation:

- IBM 4202 Proprinter XL*
- IBM Pageprinter II*
- Apple** LaserWriter Plus**.

These printers are connected to different printer ports:

- The IBM 4202 Proprinter is connected to LPT1.
- The IBM Pageprinter II is connected to LPT2.
- The Apple LaserWriter Plus is connected to COM1.

COM1 has been defined as follows:

Baud rate 9600
Parity None
Data bits 7
Stop bits 1
Handshake Hardware

Five local queues are installed:

- IBMNULL1: uses the IBM4202 Proprinter with the IBMNULL driver on LPT1.
- IBM42XX1: uses the IBM4202 Proprinter with the IBM42XX driver on LPT1.
- PSCRIPT2: uses the IBM Pageprinter with the PSCRIPT driver on LPT2.
- PAGEP__Q: uses the IBM4202 Proprinter with the IBM42XX driver on LPT1 or the IBM Pageprinter with the PSCRIPT driver on LPT2.
- LASERP_Q: uses the Apple LaserWriter Plus with the PSCRIPT driver on COM1.

One remote queue is installed:

- LANPOSTQ uses IBM Personal Page Printer II-31 with the PSCRIPT driver on the LAN Server LS:LANSRV2LANPOSTQ.

Predefined printer and job properties are used:

- Queue IBM42XX1 on printer PRINTER1 uses properties defined in the file IBM42XX.PJP.
- Queue PSCRIPT2 on printer PSCRIPT2 uses properties defined in the file PPSCRIPT.PJP.
- Queue LASERP_Q on printer PSCRIPT1 uses properties defined in the file APSCRIPT.PJP.
- Network queue LANPOSTQ uses properties defined in PPSCRIPT.PJP.

The default OS/2 queue and printer are PSCRIPT2.

The three connected printers also have their Windows equivalent installed:

- IBM Proprinter XL on port LPT1.OS2
- IBM Personal Page Printer II-031 on port LPT2.OS2
- Apple LaserWriter Plus on port COM1:

View Figure 49 on page 236 for a graphical representation of the OS/2 part of the above-described scenario. Figure 50 on page 237 shows the WIN-OS/2 part of the above described scenario.

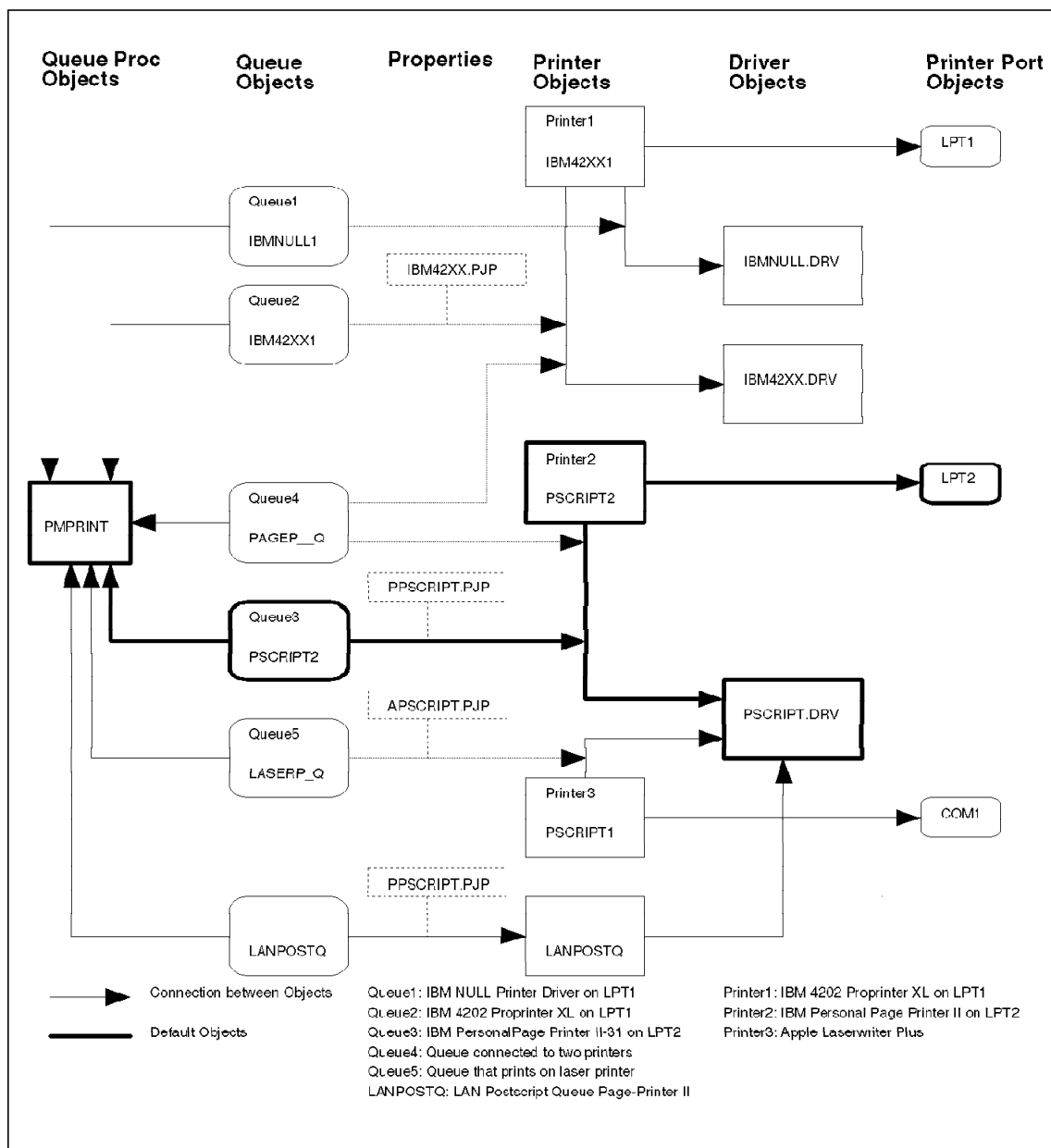


Figure 49. Sample Workstation Printer Scenario (OS/2 Part)

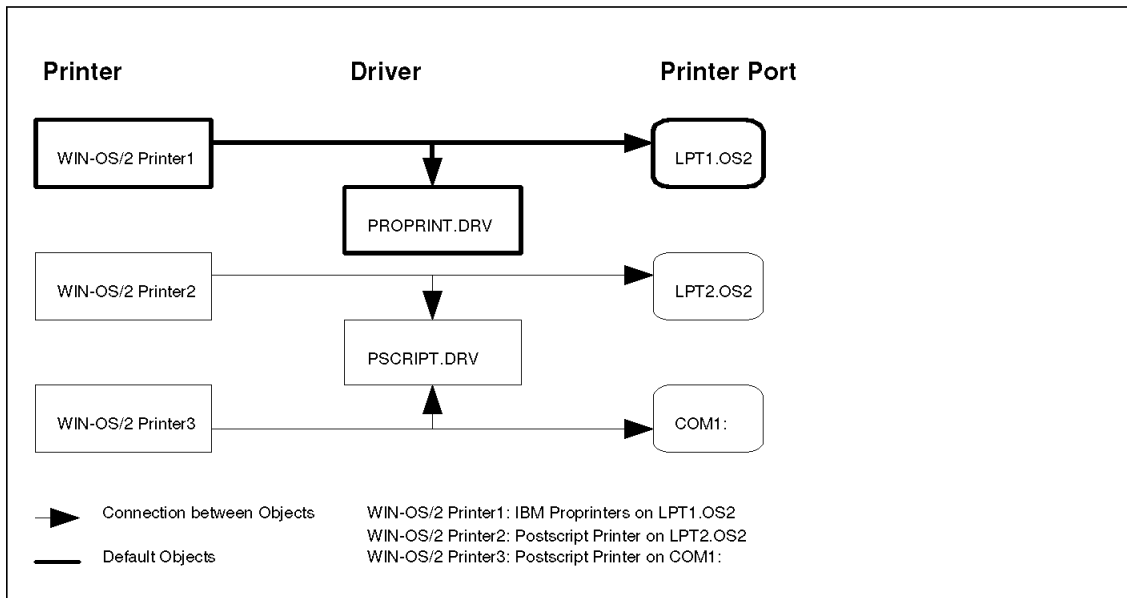


Figure 50. Sample Workstation Printer Scenario (WIN-OS/2 Part)

Figure 51 on page 238 shows the printer response file used to describe this installation.

The numbers for the printer drivers apply to a particular version of PRDESC.LST. Check the version you are using to ensure that the correct drivers are being installed.

```

Printer1 = 141 164
* 141 = IBM 4202 Proprinter XL (IBM42XX.DRV)
* 164 = IBM NULL Printer Driver
Printer2 = 166
* 166 = IBM Personal Page Printer II-31 (PSCRIPT.DRV)
Printer3 = 9
* 9 = Apple LaserWriter Plus v42_2 (PSCRIPT.DRV)
*
PrinterPort1 = LPT1
PrinterPort2 = LPT2
PrinterPort3 = COM1
*
PrinterDesc3=Apple Laserwriter Plus
*
Queue1 = 1,2
* Queue1 connected to Printer1 and uses IBMNULL
Queue2 = 1,1
* Queue2 connected to Printer1 and uses IBM42XX
Queue3 = 2
* Queue3 connected to Printer2 and uses PSCRIPT
Queue4 = 1,2
* Queue4 connected to Printer1 and uses IBMNULL
Queue4 = 2
* Queue4 also connected to Printer2
Queue5 = 3
* Queue5 connected to Printer3 and uses PSCRIPT
*
QueueName4 = PAGEP__Q
QueueName5 = LASERP_Q
*
QueueDesc4 = Queue connected to two printers
QueueDesc5 = Queue that prints on laser printer
*
Properties=1, 2, Z:\IBM42XX.PJP
Properties=2, 3, Z:\PPSCRIPT.PJP
Properties=3, 5, Z:\APSCRIPT.PJP
*
NetQueue= LS:\\LANSRV2\LANPOSTQ, 131, "LAN Postscript Queue Page-Printer II",
PMPRINT, Z:\PPSCRIPT.PJP
* IBM Personal Page Printer II: IBM Personal Page Printer II-31 (PSCRIPT.DRV)
*
DefaultPrinter = 2
*
DefaultQueue = 3
*
WinPrinter="IBM Proprinter XL **", LPT1.OS2
WinPrinter=2, 3
*
AdditionalPrinter = 43 124
* 43 = Epson LQ-1050 (N9) 24 pins - 136 columns: (EPSON.DRV)
* 124 = IBM 4029 Laserprinter 10L (LASERJET.DRV)
*
CommPort1=9600,N,8,1,H

```

Figure 51. Sample Printer Response File

7.6.1 Response File Configurator

This sample application helps the administrator to create the RMPI response file using the PM front-end application instead of editing a text file. The configurator can be found on the sample code diskette.

Following is a short description of the configurator (RMPI_CFG.EXE):

RMPI_CFG [responsefile] [/DSC:printerlist] [/W PR:winprinterlist]

Parameters:

responsefile	Name of a response file that should be loaded on program start. If the file doesn't exist or does not contain valid response file data, no message will appear.
printerlist	If the PRDESC.LST is not in the actual directory, the drive, path and file name of the PRDESC.LST has to be defined with this parameter.
winprinterlist	If the file CONTROL.INF from WINOS2 is not in the actual directory or if another Windows printer list should be used, the drive, path and filename can be defined with this parameter.

The parameters can be used in any order.

Some items in the pull-down menus are not available in some cases:

- To define a New Printer Queue is only possible if a printer is defined first.
- New Printer/Job properties is only selectable if at least one printer and one queue are defined.
- Setting Defaults is only available if at least one printer is defined.
- Change Definition or Delete is only available if a printer, queue, PJP definition, Netqueue or Windows printer definition is selected.
- Additional Connection of Queue is only selectable if a queue is selected and another printer is available to which the queue is not connected yet.
- Save is only available if the actual response file has a name already. That means a response file was loaded or the response file was saved via Save as before.
- Save and Save as are not available if there was no change since loading or saving of the response file.

If the program detects invalid definitions in a response file that is loaded, the invalid lines or definitions are ignored.

7.6.2 Printer Installation Sample

This program can be used in three ways. It can be started in an OS/2 window or full-screen session using Drive A: or B:, or any other local or LAN drive. If a drive letter other than A: or B: is used, the program looks for root subdirectories PMDD_1 to PMDD_n on that drive.

It also can be used during the regular remote install procedure. In this case it only needs one entry in the response file to trigger the execution of the program. A sample of this would be:

```
UserExit=printers.cmd
```

This sample line calls a CMD procedure called PRINTERS.CMD. This can be located anywhere along the PATH statement as set in the CONFIG.SYS. The contents of PRINTERS.CMD could be:

```
REM PRINTERS.CMD
REM set the correct working directory,
C:
CD \OS2\DLL
REM call the program
X:\EXE\RINSTPRN /R:X:\RSP\RMPI\PRINTER.RSP /DSC:X:\IMG\OS2V21\PMDD_1\PRDESC.LST
/DRV:X:\IMG\OS2V21\PMDD_1\PRDRV.LST /S:X:\IMG\OS2V21
/T:C: /WT:C: /L1:N:\RINSTPRN.LOG /WPR:X:\EXE\OS2V21\CONTROL.INF
/WDR:X:\EXE\OS2V21\DRVMAP.INF
```

Figure 52. PRINTERS.CMD to Be Called from UserExit

Note: For RINSTPRN to run properly, it needs access to the C:OS2DLL subdirectory, which was installed by the RSPINST program before it executed the UserExit keyword. RINSTPRN needs access to some DLLs. The LIBPATH statement must therefore be set up exactly as shown below in the CONFIG.SYS sample.

If we assume that the X: drive is the redirected drive, on which the PRDRV.LST and PRDESC.LST reside on the PMDD_1 root subdirectory, the program would read these two files from the specified drive.

The PRINTER.RSP response file for remote printer install is also assumed to reside on a redirected drive. In this sample it resides on the redirected X: drive. It is also possible for each workstation to use its own customized

PRINTER.RSP file. For more details see the advanced scenarios for the different client/server environments described in this document.

The results (in a log file) are stored on the redirected N: drive, to which the client must have write access.

The LIBPATH statement *must* have the “.” as first entry to have OS/2 look at the working directory for DLLs.

7.7 Using LCU

The third method utilizes the LCU and NVDM/2. The RMPI application has to be defined as a product in the LCU command file in the product definition section and also entered to the LCU command file queue for execution. For distribution using NVDM/2 a RMPI profile has to be created.

The application can be executed in the LCU command file queue with other products without rebooting or in the co-req group when NVDM/2 is used.

A sample LCU command file and NVDM/2 RMPI profile can be found on the sample code CDROM.

Chapter 8. Auto-Partitioning the Hard Disk

This chapter provides information on the Fixed Disk Utility program and on the sample applications automating the partitioning of a hard disk.

8.1 Multiple Fixed Disks

The following table shows the hardware specifications when using multiple fixed disks, under OS/2 Warp V3.

<i>Table 9. Hardware Specifications</i>	
	OS/2 Warp System Max.
Fixed Disk Drives Supported	24
Logical Drives Supported	26
Maximum Disk Partition Size	> 2 G B
Maxi.# of Primary Part./Drive	4
Maxi.# of Primary Part. w/ an Extended Part.	3
Maximum # of Partitions/Disk	24
Diskette Drives Supported	3

8.1.1 Restrictions

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 32MB.
- **Boot Manager:** Must reside in the first 1024 cylinders of the first hard disk. Typically the first 1024 cylinders is equal to 1GB or 1024 MB.

The Bootmanager is supported by the FDISK program shipped with OS/2 V2.0 and later. If the harddisk is already partitioned you have to free space by deleting a partition for the new Bootmanager partition. This destroys the data located on the part of the disk that is to be repartitioned.

8.2 The Fixed Disk Utility Program (FDISK)

The Fixed Disk Utility Program (FDISK) 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 for OS/2 or DOS utilizing its own user interface and callable from a batch file (CMD).
This program has a full screen non-PM interface because it is used in several environments where PM is not available.
- FDISKPM for OS/2 under the control of Presentation Manager.
It is used to update disk environments on live operating systems.

8.2.1 Description of FDISK for OS/2

FDISK for OS/2 enables the user to partition the hard disks and specify Boot Manager support options.

The following figure is discussed in detail below.

FDISK				
Disk 1 2				
Partition Information		Access	FS Type	MBytes
Name	Status			
	Startable	: Primary	BOOT MANAGER	1
OS2WARP	Bootable	C: Primary	FAT	130
RESIDENT	Bootable	D: Logical	HPFS	72
F1=Help F3=Exit Tab=Disk Enter=Options Menu				

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. 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. This name is specified using the "Add to Boot Manager" Menu option and can be changed by using the "Change Partition Name" option.

- **Status**

Indicates if a partition is **Installable**, **Bootable**, **Startable** or none of the above.

If set **Installable**, the respective partition will be used as the target for continuing the OS/2 install.

If set **Bootable**, the respective partition is displayed on the Boot Manager menu when the system is restarted.

If set **Startable**, the system restarts directly from this partition and will be **Installable**.

Remember: One of the primary partitions must be set Startable for the system to restart successfully.

- **Access**

Specifies if a partition is accessible. The letter in the column indicates that the partition is accessible. This column also indicates if the partition is a primary partition or a logical drive within the extended partition. Only one primary partition is accessible at a time on each physical drive. So if you want to access the data of the DOS partition from the OS/2 system, install OS/2 in a logical drive in the extended partition.

- **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 to display the Options pull-down.

Use the tab key to activate the disk selection.

8.2.2 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 secondary partition, or logical drive.
- Add to Boot Manager menu...
New partition bootable, selectable from 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.

These startup values can be set using the SETBOOT program also. For a detailed description of SETBOOT refer to the OS/2 Command Reference.
- Remove from Boot Manager menu
- Delete Partition
- Set Installable
This partition is ready to receive a new operating system.
- 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.

All of the functions which are updating the size or the location of a partition force a reboot.

For more information about FDISK see *OS/2 Warp Connect Users Guide*.

8.3 FDISK Command Line Interface

In order to modify logical drives, change Boot Manager options via batch files or remotely controlled interfaces like RCF for unattended environments, you can use the FDISK program with command line parameters. A command line interface is needed to provide the functions performed by FDISK in the PM and the full screen install environment.

Note

This section is not intended to be a complete reference for the FDISK command, merely a summary of some important parameters. The FDISK command is fully documented in the *Online OS/2 Command Reference*.

The basic FDISK command line reads as follows:

FDISK /command:value /restrictor:value

Each FDISK statement consists of a:

- Command, with or without a value and
- Restrictors with or without values

which are described in the following section.

8.3.1 The Command Parameter of the FDISK Command Line

The command parameter initiates the actual execution of the command. The following *command:values* are available:

- **/query**
Returns a list of all partitions and unused areas on the disk(s).
- **/create:name**
Creates a partition with the optional boot *name* assigned.
- **/delete**
Deletes a partition. To delete all partitions on a physical disk use **/delete:all /disk:n** where *n* is the disk number.

Note

Be careful using the **/delete:all** parameter. In the newer PS/2 models the reference partition with the system hardware configuration data is no longer hidden. So it will be **deleted** !! You can specify a filesystem type in the restrictor parameter FSTYPE to avoid the deletion of the reference partition.

- **/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 bootable from the Boot Manager menu.
- **/setaccess**
Sets accessibility of partition (creates the drive letter). Use this setting to select a primary partition as the C: drive if you have multiple primary partitions.
- **/startable**
Sets a partition startable, thereby making it the default partition to start from automatically. The OS/2 installation process will automatically set the Boot Manager partition Startable, if one is present.
- **/file:filename**
Processes all FDISK commands in the file *filename*. The restrictors must be separated from the command and each other by commas. See 8.5.1, "The FDISK 'FILE:' Parameter" on page 252 for an example of the use of this command.

Before using the /FILE parameter the BOOTMANAGER-Partition must be created.

8.3.2 The Restrictor Parameter of the FDISK Command Line

Contrary to commands, restrictors are arguments which 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 "/drive:2 /vtype:1" is added, then only the primary partitions on drive 2 would be output.

The following restrictors and associated values are available:

- **/name:ccccccc**
Specifies a partition boot *name*. The value *ccccccc* may be any alphanumeric, special character, or blank and is case sensitive and must be enclosed in parenthesis if imbedded blanks are used.

Example:

/name:"Sys OS2".

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 only used as a temporary identifier for the user. Since the user doesn't have a visual representation as with the full screen FDISK, these pseudo names can be used in place of real names for the name restrictor.

- **/disk:n**
Specifies the disk number. The value of *n* can be any number between 1 and the number of disks in the workstation.
- **/fstype:hxx (or) :ttttt**
File system type:
hxx = where *xx* is the system indicator as defined in the partition table
or *ttttt* can be:
 - dos
 - fat
 - hpfs
 - free
 - other
- **/start:c**
Create partition starting at location *c*, where *c* can be:
 - t = top of partition
 - b = bottom of partition
- **/size:mmmmmm**
The size of the partition where *mmmmmm* is the amount of space in megabytes.
- **/vtype:X**
Specifies the partition type. The value of *X* can be:
 - 0 = unusable
 - 1 = primary
 - 2 = logical
 - 3 = primary or logical
- **/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

8.3.3 Output Created by FDISK

The command line FDISK will return a return code and the requested query information which can be piped and/or redirected. The return codes are not completely documented. There are some return codes other than '0' that describe a successful operation. We have tested a lot of fdisk operations to get a mostly complete list of the return codes:

- 0 for a successful operation and
- 1 for an unsuccessful operation
- 5120 (decimal) or 1400 (hex) Successful operation
- 7168 (decimal) or 1C00 (hex) - Successfully created Bootmanager partition.
- 7680 (decimal) or 1E00 (hex) - Successfully deleted all partitions of given VTYPE
- 6144 (decimal) or 1800 (hex) - Successfully deleted Bootmanager partition.

If you only look at the second byte of the return codes, it is true that a successful program operation is indicated with return code '00'.

The output shown below is the result of the following statement:

FDISK /query

Drive Name	Partition	Vtype	FStype	Status	Start	Size
1 00000020	:	1	0a	0	0	1
1 DOS	C:	1	01	1	1	5
1 OS2V2	D:	2	07	1	6	40
1 DATA	E:	2	01	1	46	12

Figure 53. Sample Output from FDISK Command Line Query

8.4 The OS/2 SETBOOT Command

If Boot Manager is installed on the system the facilities of the OS/2 SETBOOT command also become available. SETBOOT is documented in the online *OS/2 Command Reference*.

The following commands may be useful to enhance automation of client fixed disk partitioning:

- This command sets the default partition to be booted as "OS2V3":
`SETBOOT /0:OS2V3`
- These commands set the partition to be booted on the next IPL as "SEED30":
`SETBOOT /1:SEED30`
`SETBOOT /X:1`
- This command reboots the workstation:
`SETBOOT /B`
- This command immediately reboots the workstation using the partition which is the E: drive:
`SETBOOT /IBD:E`

The SETBOOT /IBD:C command does *not* require Boot Manager to be installed to operate. This command is used in DISKPREP.CMD to reboot the workstation without user involvement.

8.5 The Sample REXX Partitioning Utilities

Two programs and several RSP files are included:

- FDSKHD1.RSP and FDSKHD2.RSP
FDSKHD1.RSP is a responsefile specifying OS/2 FDISK functions for a system with one physical Harddisk. FDSKHD2.RSP is a responsefile for a system with two physical Harddrives.
- PIPE.CMD
PIPE.CMD is a utility that takes the output from a command and places it onto the REXX queue.
- DISKPREP.CMD
DISKPREP.CMD performs the actual disk partitioning.

The .CMD files mentioned above have to be copied to the CID\EXE subdirectory and the .RSP files to the CIDRSP subdirectory of the recommended directory structure.

8.5.1 The FDISK 'FILE:' Parameter

8.3, "FDISK Command Line Interface" on page 247 describes all valid FDISK command line parameters. DISKPREP.CMD is written so that all disk partitioning is done with one command:

```
FDISK /file:FDSK80.DAT
```

This allows the administrator to define how the fixed disk will be partitioned independent of the DISKPREP.CMD program. The example below creates a drive setup with three partitions, assuming Boot Manager has already been installed:

```
/create:OS/2,/vtype:2,/size:200,/disk:1,/start:t  
/create:SERVICE,/vtype:2,/size:20,/disk:1,/start:t  
/create,/vtype:2,/disk:1,/start:t
```

Figure 54. Example FDSK.RSP File

8.5.2 PIPE.CMD

The PIPE.CMD should be placed in the CID\EXE directory also. PIPE.CMD is included on the sample code diskette.

```
/* Take lines piped in and queue them */  
do while lines()  
  line=linein()  
  queue line  
end
```

Figure 55. PIPE.CMD Program

8.6 Using REXX Code to Create Partitions on Hard Disks

The default installation using a response file creates one large partition. In order to achieve the flexibility required to automate the fixed disk partitioning the power of REXX is needed.

It is assumed that the sample code (DISKPREP.CMD) will be executed before the OS/2 response file installation is started. This code assumes that the administrator wishes to re-partition the client fixed disks as part of the installation process.

All data on the client fixed disks will be lost.

A backup procedure should be included as part of the installation procedure if there is data on the client disks which must be saved. This data could be restored in a *UserExit* of the response file installation.

8.6.1 Accessing REXX from a Client Workstation

In order to gain access to OS/2 REXX from the minimal systems booted for installation, several requirements must be met:

1. The OS/2 REXX DLLs must be in a directory that is referenced in the LIBPATH.
2. The OS/2 REXX MSG files must be in a directory that is referenced in the DPATH.
3. The program SRVREXX.EXE must be run to initialize the REXX functions.
4. If you use the MPTS-LCU, you can use the CASCKREX.CMD program to wait for REXX support to be initialized completely. This is useful because the initialization of REXX can take several seconds.

The REXX programs DISKPREP.CMD and PIPE.CMD should be copied to the CID server CID\EXE subdirectory along with FDSK*.RSP response files, so they become accessible from the client's redirected drive. The programs are explained in detail below.

The FDSK*.RSP files that define how the client drives will be partitioned must also be copied to the code server. See 8.5.1, "The FDISK 'FILE:' Parameter" on page 252 for a description of FDSK*.RSP.

8.6.2 Summary of DISKPREP.CMD Function

The DISKPREP.CMD Function requires the following invocation parameters:

- /R:Response file path - Fully qualified path for the FDSKHD*.RSP files.
- /E:ExePath - Fully qualified path for PIPE.CMD and SETBOOT.EXE
- /S:SourcePath - Source Directory for the OS/2 Image
- /L:Logfile - Fully qualified filename for the logfile.
- /F:Format Flag - Possible values 'Y' or 'N'. Default is 'Y' for formatting the volumes.

The sample REXX code will perform the following steps:

- Check for the existence of a partition named "OS/2." If such a partition exists the function FIRSTFORMAT() is called to format all found volumes with the HPFS filesystem. The formatting of the volumes is integrated in this batch to avoid the quick format of the OS/2 Warp V3 installation process. You can suppress the formatting by using the /F:N parameter.
- If no such partition exists, the code will delete all partitions on the disk and run the FDISK program to create the partitions required.
- The disk is partitioned using the response file capability of the FDISK command (the "/file:" parameter). Based on the number of physical hard disks in the system, the file FDSKHD1.RSP or FDSKHD2.RSP is used to create the partitions needed.

You can modify the *.RSP files to fit to your installation concept.

- The user will then be prompted with a panel to reinsert the "Installation" diskette and reboot.
- The second time through the procedure, an "OS/2" partition will exist so if required the formatting is done and installation will continue.

8.6.2.1 Subroutines

The following shows a short description of the subroutines use in the DISKPREP.CMD program.

- NAMECHECK: Checks for the existence of a partition named "OS/2".
- Help: Shows the Syntax and program invocation parameters of DISKPREP.CMD.
- GetFDiskLine: Scans one line from the output of an FDISK /query command and fills the variables with the values in this line.

- CheckDrives: Checks for the number of physical Harddisks and returns the number in the variable maxdrives. The number of volumes and the names are returned in the variable Vol..
- DELETEPART: Main function for deleting of existing partitions and creating the new partition table.
- FIRSTFORMAT: Formats all logical Volumes with the HPFS file system.
- PrettyBox: Draws a message box.
- CIDInit: Initializes the CID return code variables.
- Log: Adds a line to the logfile.
- Cmd: Executes an OS/2 command adding the errors to the logfile.
- KillQueue: Deletes all remaining entries from the queue.

For a complete Listing of the DISKPREP.CMD see Appendix M, "DISKPREP.CMD" on page 617

8.7 Disk Partitioning when Using NVDM/2

When you are using NetView DM/2 V2.1 as the software distribution manager the CC client connected to the CC server with the boot diskettes does **NOT** have any REXX support loaded. If you want the client to execute REXX-programs you have to install REXX support on the client. We will describe here what we have done to implement this.

1. Copy the necessary files for REXX-support to the NetView DM/2 V2.1 Server into the SHAREADLLCONNECT directory. For a detailed description of how to install the REXX-Support refer to 17.1.2, "GETREXX" on page 398.
2. The OS/2 REXX DLLs must be in a directory that is referenced in the LIBPATH.
3. The OS/2 REXX MSG files must be in a directory referenced in the DPATH.
4. You have to create a change file profile for the REXX-Support.

REXX-change file profile

```
TargetDir=C:\

Section Catalog
Begin
    ObjectType = SOFTWARE
    GlobalName = UTIL.REXXSTART.300.INST.REF.1
    Description = REXX-Support for Pristine Clients
End
; This Changefile has to be installed together with the DISKPREP-
; changefile in a corequisite-group.
; Function: Detaches the SRVREXX.EXE Program shipped with MPTS=LCU
; to activate the REXX-Support.

Section Install
Begin
    Program      = SA:\EXE\CONNECT\CMD.EXE
    Pams = /c $(SA)\EXE\CONNECT\DETREXX.CMD $(SourceDir) $(WorkingDir)
    SourceDir    = SA:\DLL\CONNECT
    WorkingDir   = SA:\DLL\CONNECT
End
```

5. You have to add the referenced BATCH-file to your EXECONNECT subdirectory. The batch file is used to detach the SRVREXX.EXE command and to wait for the REXX support to be initialized.

DETREXX.CMD

```
@goto begin

:begin
SET PATH=%PATH%;%2;
set helper=%1\SRVREXX.exe

if not exist %helper% goto error
detach %helper%
call %2\casckrex
goto end

:error
@echo .
@echo %helper% not found. NDM-Server updated for REXX-Support ?
@cmd
@goto end

:end
```

6. Create a change file profile for the DISKPREP.CMD

```
DISKPREP.CMD change file profile

TargetDir=C:\

Section Catalog
Begin
  ObjectType = SOFTWARE
  GlobalName = IBM.OS2.CONNECT.DISKPREP.INST.REF.1
  Description = Automated Partitioning for OS/2 PCs
End

Section Install
Begin
  Program = SA:\EXE\CONNECT\DISKPREP.CMD
  Pargs = /R:$(SA)\RSP\CONNECT /E:$(WorkingDir) /S:$(SourceDir) /L:$(LogFile1) /F:Y
  SourceDir = SA:\IMG\CONNECT
  WorkingDir = SA:\EXE\CONNECT
  LogFile1 = SB:\LOG\DISKPREP\$(WorkStatName)\FDISK.LOG
End
```

7. Now you can install the REXX support and the DISKPREP.CMD program as a **corequisite group**. It is necessary to install them as a corequisite group to have the REXX support loaded again after the reboot.

There is a detailed description of how to add REXX support for a pristine client included in *OS/2 System Software Distribution & Installation Using NetView DM/2*, GG66-3253.

8.7.1 Write a Batch Procedure without REXX for NVDM/2

If you have a common partition for all or at least most of your systems and no need to query for the actual status of the disk you could also use an OS/2 command procedure only executing FDISK and SETBOOT without using REXX. The following shows an example for the partitioning using only FDISK and SETBOOT:

```
FDISK /delete:all /disk:1 /FSTYPE:HPFS >NUL;
FDISK /delete:all /disk:1 /FSTYPE:FAT >NUL;
FDISK /delete:all /disk:1 /FSTYPE:DOS >NUL;
FDISK /delete:all /disk:1 /FSTYPE:HOA >NUL;
FDISK /create /disk:1 /bootmgr /start:t >NUL
FDISK /create:OS2 /size:99 /vtype:1 /disk:1 /bootable:1 >NUL
FDISK /create:DATA /size:264 /vtype:2 /disk:1 /bootable:0 >NUL
FDISK /create:MAINT /size:15 /vtype:2 /disk:1 /bootable:1 >NUL
FDISK /starttable /name:OS2 >NUL
SETBOOT /T:0
SETBOOT /IBD:C:
```

Make sure you have FDISK.EXE and SETBOOT.EXE in the specified directory.

If you want to work with input files instead of system-specific procedures, you may want to use the /FILE parameter of the FDISK command.

Following the standard for client workstations in the previous scenarios, the pristine workstation can be partitioned in the following way:

- C: drive - primary partition (100MB)
- D: drive - extended partition (200MB)

For example, the following PREPWS.CMD procedure can be used:

PREPWS.CMD Procedure

```
@echo off
if exist a:\disk1.dat goto step2
echo *****
echo **
echo ** Step 1: Partitioning
echo **
echo **           Please wait ....
echo **
echo *****
fdisk /file:FDISKD.DAT
fdisk /create           /disk:1 /bootmgr    /start:t >NUL
fdisk /file:FDISKN.DAT
echo Partitioning done > a:\disk1.dat
ECHO Please insert diskette 1/2
pause
SETBOOT /T:10
SETBOOT /IBD:C:
:step2
echo *****
echo **
echo ** Step 2: Formatting
echo **
echo **           Please wait ....
echo **
echo *****
echo Y | format c: /FS:HPFS >NUL
echo Y | format d: /FS:HPFS >NUL
echo Y | format e: /FS:HPFS >NUL
del a:\disk1.dat
```

This procedure first deletes existing partitions with a separate input file called FDISKD.DAT.

FDISKD.DAT for PREPWS.CMD

```
/delete:all,/disk:1,/FSTYPE:HPFS  
/delete:all,/disk:1,/FSTYPE:FAT  
/delete:all,/disk:1,/FSTYPE:DOS
```

If you do not have bootmanager available on the system, you cannot assign partition names for the bootmanager menu using the /FILE parameter. After deleting the existing partitions, the procedure creates the new partitions using another input file called FDISKN.DAT.

FDISKN.DAT for PREPWS.CMD

```
/create,/disk:1,/size:100,/vtype:1  
/create,/disk:1,/size:100,/vtype:2
```

After the partitioning is done, the formatting starts. The procedure is controlled by a file mark that is written to the diskette after the first step. When the system reboots from the diskettes after partitioning, it will jump to the second step which performs the formatting.

In the input files the parameters used are as follows:

- | | |
|-------------------|--|
| /delete | Deletes all partitions on a physical disk. In the /disk:n specification, parameter n represents the disk number. The /FSTYPE restrictor specifies the filesystem type of the partition. It is specified to avoid the deletion of a reference partition on PS/2 systems with non-hidden reference partitions. |
| /create | Creates a partition. If you use Boot Manager you can also specify a boot name (/create:name). |
| /size:nnnn | Specifies the size of the partition in MB. |
| /vtype:X | Specifies the partition type. The value of X can be: <ul style="list-style-type: none">- 0 = unusable- 1 = primary- 2 = logical- 3 = primary or logical |

The following files have to be reachable on the system, meaning their path has to be added to PATH, DPATH and LIBPATH of the CONFIG.SYS on the second boot disk. We put them to the subdir \EXE\CONNECT.

- FDISK.COM - copied from C:OS2
- FDISKN.DAT - used by FDISK in PREPWS.CMD
- FDISKD.DAT - used by FDISK in PREPWS.CMD
- FORMAT.COM - copied from C:OS2
- PREPWS.CMD - OS/2 Procedure
- SETBOOT.EXE - copied from C:OS2
- UHPFS.DLL - copied from C:OS2DLL

Use the following profile to create a change file for PREPWS.CMD on the CC server.

Change file profile for PREPWS.CMD

```
TargetDir=C:\

Section Catalog
Begin
  ObjectType = SOFTWARE
  GlobalName = IBM.OS2.CONNECT.PREPWS.INST.REF.1
  Description = Automated Partitioning for OS/2 PCs
End

Section Install
Begin
  Program = SA:\EXE\CONNECT\PREPWS.CMD
  WorkingDir = SA:\EXE\CONNECT
End
```

Part 3. CID System Generation and Administration

Part three is intended for the administrator responsible for constructing the CID system. This is the person responsible for building the CID code server(s) with the LAN transport system and all source images.

This part will describe the manual preparation of the CID code server and client workstations without using CASSETUP.

This utility is described in Chapter 18, "Automated Setup with CASSETUP" on page 403.

This section describes how to install and configure a CID code server manually for the distribution of OS/2 Warp Connect, MPTS, IBM Communications Manager/2 Version 1.11, DATABASE 2 for OS/2 and the appropriate LAN requester for the environment.

Five different types of code servers will be covered: IBM Operating System/2 Local Area Network Server V5.0 RIPL, LAN CID Utility, Novell NetWare Version 4.01, IBM NetView Distribution Manager/2 Version 2.1 and IBM TCP/IP Version 3.0.

Chapter 9. Hardware and Software Requirements

9.1 Hardware

9.1.1 Server: Base Hardware

For the code server we recommend using a machine that features

- a faster bus system than the "standard" AT/ISA bus, such as:
 - Micro Channel Architecture (MCA bus)
 - Local Bus concepts like
 - PCI bus, which has become a kind of standard today.
 - VESA Local bus (VL bus), which is widely replaced by PCI today.
- at least an 80486DX processor.
- at least 16MB RAM.
- If you are using the code server for other applications running at the same time, you should consider increasing memory.

9.1.2 Server: Hard Disk Drives

In the code server, if possible, have

- Two physical hard disks.
 - One for OS/2 and the code server's base code.
 - The second disk for the CID directories.
- Since most of the time files are read from the disk, it is important to use a fast disk.
- We also recommend that the disk used for the CID directories is formatted with HPFS.
- Ensure that disk caching is enabled.
- 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 where the clients need write access to. It would also be the only disk where you will not know in advance how much space is needed.

For the requirements for different products see the *Planning and Information* manuals for the products and remember to check the README files for updates.

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.

Please remember that you will need free space on the disk(s) to hold the clients' log files as well.

The following table is only meant as a rough guideline.

<i>Table 10 (Page 1 of 2). Disk Space Recommendations for Diskette Images</i>	
Product Name	Space Needed for Images (may be rounded up)
OS/2 Warp V3 without Windows 1	36.5MB
OS/2 Warp V3 with WinOS2 support	44MB
OS/2 Warp Connect with WinOS2 support	53MB
OS/2 Warp Connect FixPak 17	20.5MB
CM/2 V1.11	11.5MB
CM Server V4.0	22MB
DB2/2 V2.11 Single-User Version	25MB
DB2/2 V2.11 Server Version	24MB
DB2 Client Application Enabler/2 Version 2.11	7.5MB
LAN Server V5.0 2	23.5MB
NetView DM/2 V2.1	14.5MB
MPTS V5.0 LAPS	4.5MB
MPTS V5.0 LCU	0.4MB
MPTS V5.0 SRVIFS	0.25MB
Sample Code CDROM	1.44MB

Table 10 (Page 2 of 2). Disk Space Recommendations for Diskette Images	
Product Name	Space Needed for Images (may be rounded up)
TCP/IP V3.0	10MB
<p>Note:</p> <p>1 During installation of OS/2 Warp V3 on top of DOS and Windows the installation requires some files from the Windows diskettes.</p> <p>If more than one Windows version needs to be stored on the server do not forget to account for that disk space as well. Supported versions are Windows 3.1 and 3.11, Windows for Workgroups 3.1 and 3.11.</p> <p>2 In the LAN Server V5.0 subdirectory tree the MPTS diskettes also are copied when LAN Server V5.0 LANINST is run to create a CID structure.</p>	

The amount of storage needed when a product is installed depends on the features you choose to install. In the table above we have estimated "maximum" installs.

9.1.3 Clients

Please see the *Planning and Information* for the products you wish to install to the clients.

Minimum requirements for installing OS/2 on a diskette-initiated system are:

1. 80386SX or higher processor
2. Fixed disk(s) with enough space to install the chosen products
3. At least 20MB free space (for swapping if needed)
4. 8MB memory or more depending on product requirements

OS/2 Warp V3 will run on as low as 4MB of memory. The installation program makes an optimization dependent on the memory available in the system. Therefore if memory is removed from the system to get the performance that can be there please re-install OS/2 Warp V3 afterwards.
5. The client must be on the same logical LAN as the server

The time it takes to CID install of course depends of the products that you are installing. But the same installation runs a lot faster on a client machine with a faster processor, faster disk and the same amount of RAM than on a slower machine.

9.2 Software

9.2.1 Servers

What software you need for the basic installation of the code server depends on the type of code server.

<i>Table 11. Basic Software for Code Server</i>					
Product Name	LAN Server V5.0 RIPL	LCU SRVIFS	Novell NetWare V3.1x only (not tested for WARP)	TCP/IP	NetView DM/2 V2.1
OS/2 Warp V3	√	√	√	√	√
DOS	√		√		
MPTS and LCU	√	√	√	√	√
LAN Server V5.0	√				
NetWare			√		
TCP/IP V3.0				√	
NetView DM/2 V2.1					√
DB2/2 V2.11					√

MPTS includes LCU and SRVIFS. MPTS is delivered with LAN Server V5.0

9.2.2 Common Requirements

Before starting to set up the code server, make sure that you have:

- All diskettes (or CD-ROMs) of the products you want to install as images
- The sample code CDROM
- The MPTS diskettes
- Formatted 1.44MB diskettes for the creation of the client diskettes
 - Typically you need two diskettes.
 - For LAN Server V5.0 RIPL on the code server, you need
 - no diskette at all, if LAN adapter is RIPL enabled
 - one diskette, if no RIPL chip present on adapter

- Enough space on the server's hard disk to hold the images

9.2.3 Clients

Two boot diskettes are required for the installation of a diskette-initiated system (see the sections later in this book for the preparation of the boot diskettes).

Chapter 10. Manual Setup of IBM Operating System/2 Local Area Network Server V5.0 RIPL

This chapter describes a method of obtaining redirected drives for the installation of OS/2 Warp Connect, MPTS, CM/2 V1.11, TCP/IP V3.0, PC/3270 for OS/2 V4.1, DB2/2 V2.11 Single-User Version and LAN Requester V5.0 using the RIPL feature of LAN Server V5.0.

Considerations on Using RIPL for Installation

There are many steps involved in setting up a RIPL server to be able to install products from it. Chapter 11, "Manual Setup of LAN CID Utility" on page 293 describes how to set up a LAN CID Utility (LCU) code server, which uses SRVIFS for LAN transport. The SRVIFS code server involves fewer steps and requires less effort to set up. It can be run together with LAN Server on the same physical machine.

If all LAN workstations are equipped with LAN adapters with the RIPL chip on them then using RIPL for installation is the preferable method.

The OS/2 installation program and the installation programs of the other products to be installed will only execute on an OS/2 system. This means that the workstation on which OS/2 is to be installed must boot a usable OS/2 system. If you wish to install from a server running LAN Server V5.0, you need a way to attach a drive letter to an alias on the server. The IBM OS/2 LAN Requester components LOGON, NET START, and NET USE require Presentation Manager to be available as well as many executable files. It is impossible to boot an OS/2 LAN Requester system from a few diskettes.

This problem was solved by using the Remote Initial Program Load (RIPL) feature of LAN Server V5.0. RIPL allows a requester to boot from an OS/2 directory structure on the server. The client machine sees its boot drive (**Z:**, corresponding to an area on the LAN Server) as well as the local drive(s). Since the system has booted OS/2, and has access to both redirected drives to a server and the local drive, the CID installation process can be used to install OS/2 V2.11, OS/2 Warp Connect, MPTS, PC/3270 for OS/2 V4.1, CM Server V4.0, DB2/2 V2.11 LAN Requester V5.0, CM/2 V1.1 and LAN Server V5.0 on the local drive using the standard remote install procedure outlined in Chapter 1, "CID History, Concepts and Scenarios" on page 3.

Since we wanted to be consistent in all server environments we decided to use two aliases: one with read-only authorization for the whole CID directory structure (**X:**) and one with read/write authorization for the CIDLOG directory structure (**L:**).

10.1 Overview of Remote Initial Program Load (RIPL)

RIPL is the process of downloading IPL files from a server to a workstation in order to start (boot) the workstation. You should review *LAN Server V5.0 Network Administrator Reference Volume 3: Network Administrator Tasks*,

RIPL can be used to boot OS/2 on a workstation with DOS or OS/2 installed on the fixed disk, on a machine with an unformatted fixed disk, or a machine with no fixed disk at all. Normally to enable a workstation to RIPL from a server, the LAN adapter in the workstation must be enabled with a RIPL ROM module, and the type of adapter must be supported by IBM LAN Server RIPL.

Important Note!

A RIPL ROM module is *not* required for installation if you are using an IBM Token-Ring LAN, an IBM Ethernet LAN or an IBM PC Network LAN. LAN Server V5.0 includes a productivity aid called MKRDPM. This utility will build a bootable diskette which simulates the code in the RIPL ROM module.

The RIPL ROM works by adding itself to a machine's boot sequence. A workstation with RIPL capability will *normally* attempt to IPL in one of the following ways:

1. From diskette if a bootable diskette is in the drive
2. From fixed disk if the drive is bootable
3. From the RIPL server

Note: Some PS/2 systems allow the user to redefine this boot sequence through the use of the reference diskette/partition.

If a workstation reaches step 3, or if it was booted with a **MKRDPM** diskette in step 1, it will broadcast a RIPL request on the LAN. A LAN Server V5.0 server with the RemoteBoot service active will respond if the workstation's LAN adapter address has been defined to that server. From this point on the

workstation will perform a normal OS/2 IPL from a subdirectory structure on the server.

The domain name used in the examples is **CIDDOM** and the code server name is **LSCIDSRV**.

10.2 Overview of Installation Steps for RIPL

In the rest of this chapter we will walk you through the manual installation of a code server using IBM Operating System/2 Local Area Network Server V5.0 with RIPL to install OS/2 Warp Connect, MPTS, CM/2 V1.11, PC/3270 for OS/2 V4.1, CM Server V4.0 LAN Requester V5.0, DB2/2 V2.11 Single-User Version and LAN Server V5.0.

In the following sections all steps will be explained in detail.

1. First of all OS/2 Warp Connect needs to be installed on the code server.
2. The administrator installs IBM Multi-Protocol Transport Services
3. The administrator installs LAN Server V5.0 with OS/2 Remote IPL support on the server system.
4. The sample command files necessary for CID installation via RIPL are copied from the sample code CDROM to the CIDEXECONNECT directory and are modified if required.
5. The administrator creates the CID directory structure and loads the products' diskette images to the code server.
6. The administrator runs the RIPLINST utility to set up the directories on the server from which the clients will IPL OS/2.
7. The administrator runs the GETRPL utility to complete LAN Server V5.0 Remote IPL configuration process. GETRPL creates the group RPLGROUP, the default OS2.INI, default desktop and default access control profiles for the RIPL machines.
8. The administrator creates NET ALIASes (or NET SHAREs) and access control profiles for the CID and CIDLOG directory structures.
9. The administrator creates a NET SHARE for the CIDEXECONNECT directory.
10. The administrator creates a master workstation definition, which will be used as a model for the definitions of client workstations.

11. The administrator edits the master workstation File Index Table (FIT) file, and the master workstation CONFIG.30 file. He/she creates a master workstation STARTUP.CMD and STARTRPL.CMD file.
12. The administrator starts the RemoteBoot service on the server.
13. The administrator creates LCU command and response files for the installation of the clients. For more information see Chapter 3, "Response Files" on page 47 and 4.4, "LCU Command File" on page 143.
14. The administrator prepares a Remote IPL diskette. This is a DOS system diskette that is either prepared with the MKRDPM utility or with the RPLENABLE.EXE. Copies of the RIPL diskette are distributed to the client workstations.
15. At one test client, the RIPL diskette is inserted and the client is booted.

If the RIPL diskette was made with MKRDPM, the client will now RIPL from the server.

If the RIPL diskette used RPLENABLE, the diskette has to be removed and the workstation must reboot again. The client will now RIPL from the server using the RIPL Boot Prom on the LAN adapter.
16. The administrator creates workstation definitions for each of the client workstations to be installed, using the master workstation definition as a model. This requires entering a workstation name and the universally administered LAN adapter address for each of the client workstations.
17. The RemoteBoot service is started again and everything is ready for CID installations.

FIT Files

The concept of File Index Table (FIT) files is very important to the understanding of LAN Server V5.0 Remote IPL. If you are not familiar with the concept, review the LAN Server V5.0 documentation.

For example, the client “sees” the following mapping of drives and directories to the server:

Client drive	Server directory
Z:	IBMLANRPLUSER“client name.” For example, from the remote IPL workstation called RPCLIENT the Z: directory is IBMLANRPLUSERRPCLIENT on the server.
Z:OS2INSTALL	IBMLANRPLOS2.30OS2INSTALL

As you can see, RIPL does not use a “normal” mapping of client drives and directories to server directories. “Normal” mapping is however used for the CID and CIDLOG directory structure. In our scenario the client sees the CID directory structure as **X:** and the CIDLOG directory structure as **L:**. This was done in order to be consistent with the CID installation on the other types of code servers described in this book and which are documented in the MPTS literature.

10.3 Manual Installation of LAN Server V5.0 RIPL Code Server

From the RIPL installations we made when testing this scenario we have provided the command files on the sample code CDROM in the RIPL directory. Please note that you should look at them to determine if they need editing, especially if you are using another CID directory structure or if you are installing another OS/2 version.

10.3.1 Preparation of Basic Code on the Code Server

This section covers steps 1 - 3 of the overview. For the versions we used when testing this scenario please see Appendix B, “Versions Used in This Book” on page 431.

If you have an old code server with LAN Server V3.0 which currently is not set up to RIPL OS/2 Warp Connect you should apply Service Pak IPx7060 to it. (The x is substituted with a character corresponding to the language version of LAN Server V3.0 you are using.)

Please remember that the basic installation of OS/2 Warp Connect and LAN Server V5.0 enabled for RIPL of OS/2 Warp Connect takes about 130-150 MB of disk space. For each OS/2 RIPL machine that is defined additional space is used. We recommend that you install the base code to another drive and if possible to a physical disk other than the one where you will make the CID directory structure, especially if you intend to install more than a few clients concurrently.

1. Install OS/2 Warp Connect. If you do not have to count every byte of disk space make a “full” installation of OS/2 to make sure you are not missing anything.
2. Install IBM Multi-Protocol Transport Services program.
3. Install LAN Server V5.0 with OS/2 RIPL support. The sample names we used were CIDDOM for the domain and LSCIDSRV for the server.

10.3.2 Creating the CID Directory Structure

The recommended common CID directory structure to be used with RIPL is described in Chapter 2, “Recommended CID Directory Structure” on page 39. This common directory structure has been developed for the CID process to ensure compatibility/migration regardless of what server type will be used to provide the LAN transport and redirected drive capability. RIPL, by itself, does not require any fixed directory structure. However, we recommend the use of the CID common directory structure to be able to use the command files we provided on the sample code CDROM and to avoid compatibility problems with follow-on products that might be shipped in the future.

Ensure that the disk you want to use has enough free space to hold the desired product images and has additional space available for response, LCU command and log files. See Table 10 on page 264.

Use **MKDIR** or **MD** to create the directory structure.

When you follow the examples later in the text please remember to use the correct drive letter. The examples assume that the disk with the CID directory structure is on D:.

10.3.3 Loading OS/2 CID Utilities to Code Server

Please see Chapter 15, "OS/2 CID Utilities" on page 373 on how to load OS/2 CID Utilities to the LCU code server.

10.3.4 Copy Files from the Sample Code CDROM to Code Server

This section covers step 4 of the overview.

You can manage without using anything from the sample code CDROM, but then you have to do **everything** yourself. As seen in Part 2 of this book there are some nice utilities on the sample code CDROM. A couple of the command files originally provided with MPTS are updated to work with OS/2 Warp Connect.

Assuming that the sample code CDROM is accessed as E: enter the following commands:

```
XCOPY E:\utility D:\cid\exe\connect
XCOPY E:\sample D:\cid\img\sample
COPY E:\ripl\reqdele1.cmd D:\cid\exe\connect
COPY E:\ripl\reqd1300.cmd D:\cid\exe\connect
COPY E:\ripl\requpdat.cmd D:\cid\exe\connect
COPY E:\ripl\rmtree.cmd D:\cid\exe\connect
COPY E:\ripl\thinr300.cmd D:\cid\exe\connect
COPY E:\ripl\connect.cmd D:\cid\client
XCOPY E:\ripl\start*.cmd D:\cid\sample\ripl
```

10.3.5 Loading Diskette Images

Please see Chapter 16, "Loading Product Images to Code Server" on page 379 on how to load the product diskette images to the LAN Server V5.0 RIPL code server.

The minimum requirements when using a LAN Server V5.0 RIPL code server is that you load:

- OS/2 Warp Connect. See 16.1.1, "Loading OS/2 Diskette Images with SEIMAGE" on page 379.
- MPTS. See 16.1.2, "Loading LAN Transport System Diskette Image(s) with LAPSDISK" on page 382.
- LCU files. See 16.1.8, "Loading LAN CID Utility Files" on page 394.
- SRVIFS files. See 16.1.9, "Loading SRVIFS Files" on page 394.
- LAN Server V5.0 files. See 16.1.5, "Loading LAN Server Diskette Images" on page 386.

10.3.6 Copy REXX to Code Server

GETREXX helps you copy the REXX support necessary for the clients to be able to run their command files. See 17.1.2, "GETREXX" on page 398.

10.3.7 Copy SETBOOT.EXE and XCOPY.EXE to Code Server

GETBOOT helps you copy SETBOOT.EXE and XCOPY.EXE, which are necessary for the clients to be able to run their command files. See Chapter 17.1.1, "GETBOOT" on page 397.

10.3.8 Code Server Installation.

For a LAN Server V5.0 RIPL code server as you will see below, there are a few things that need to be done manually. They have to be done in the correct order. This section covers steps 6 - 12 of the overview.

1. To set up the OS/2 RIPL directory structure use the RIPLINST utility of OS/2. A detailed description of RIPLINST can be found on *IBM Operating System/2 Local Area Network Server V5.0 Network Administrator Reference Volume 3: Network Administrator Tasks*.

RIPLINST is OS/2 version dependent and therefore comes with OS/2 and not with LAN Server V5.0. So you have to ensure that you are using the correct RIPLINST. It has to be unpacked from the OS/2 diskettes. For OS/2 Warp Connect it is in the bundle file RIPLINST on diskette 7. The nice thing is that you can both unpack and execute RIPLINST from the OS/2 Warp Connect images you loaded to the CID directory structure.

OS/2 Warp Connect RIPLINST example:

```
CD cidexeconnect
UNPACK D:\cid\img\connect\disk_7\RIPLINST D:\cid\exe\connect
RIPLINST
Change the Source Code Directory to D:\cid\img\connect
and the OS/2 Remote IPL Directory to D:\ibmlan\rp1\connect
```

Ensure that the drive letters are correct. We recommend that you use the default directory structure which as supplied by the GETRPL command. (If you are a very experienced RIPL administrator you may be able to use a different directory. But you have to do a lot of manual editing, since there is no tool to help you.)

2. Logon as an administrator.
3. Execute the GETRPL utility.

Please read the information regarding GETRPL in the LAN Server V5.0 INF-file before executing GETRPL. How GETRPL works is described in *IBM*

Operating System/2 Local Area Network Server V5.0 Network Administrator Reference Volume 3: Network Administrator Tasks. **Note:** The LAN Server documentation comes together with OS/2 Warp Connect. You can install it using then **INSTALL.CMD** in the *BOOKINST* directory on the OS/2 Warp Connect product CD. It is the documentation for the LAN Server V4.0 but the function hasn't changed so it is still valid.

This utility creates the group RPLGROUP, the default OS2.INI, OS2SYS.INI, default Desktop and Access Control Profiles for the RIPL machines. It also updates the IBMLANRPLRPL.MAP file. This enables you to choose an appropriate "Server Record Identifier" and default FIT file, which is done later on when you are defining a "Remote IPL Machine."

4. Ensure that the REMOTEBOOT service is not running. NET START without parameters will tell you what is started. And NET STOP RPL will stop REMOTEBOOT if it is running.
5. Make the created CID directory structure accessible as a resource in the network.

This can be executed via NET ALIAS definitions or via NET SHARE statements. There is no difference between the two possibilities for our scenario, but the startup file used during the installation process has to have the corresponding statements either to the NET ALIAS (as described below) or the NET SHARE.

Define aliases via the LAN Server V5.0 GUI panels. Use *Definitions* in the main panel, *Alias*, and *Files*.

- a. Define an alias, CID, for the C:CID directory and share it as requested by user.
- b. Create an access control profile for the CID directory structure, by selecting *Access Control* in the *Alias* panel. Set the permissions for the users group to N. Select *Grouplist* and give the RPLGROUP the permissions XR for CID. Make an APPLY.
- c. Define an alias, LOG, for the C:CIDLOG subdirectory and share it as requested by user.
- d. Update the access control profile for the CIDLOG directory structure, by selecting *Access Control* in the *Alias* panel. Set the permissions for the users group to N. Select *Grouplist* and give the RPLGROUP permissions XRWCD for LOG. Make an APPLY.

Alternatively, NET SHAREs can be defined in the SRVAUTO.PRO as described below for the D:CIDEXEconnect directory. If NET

SHAREs are used do not forget to create access control profiles and to APPLY them.

- e. When you copied files from the sample code CDROM the CRENVVAR.EXE found in the UTILITY directory was copied to CIDEXEconnect. CRENVVAR prompts the user for a workstation name and puts the name in an OS2 environment variable MACHINE. The setting of MACHINE is saved in ENV_VARS.CMD, for further logons. The variable MACHINE is used for the LOGON of the client when the installation process starts. As there are several reboots in the process, the user at the client machine does not have to logon after every reboot because after the first logon ENV_VARS.CMD is used. Please refer to Appendix F, "Create Environment Variables Program Description" on page 545 for further information on CRENVVAR.EXE.
- f. One additional NET SHARE statement has to be added to the IBMLANPROFILESSRVAUTO.PRO for D:CIDEXEconnect. (This will enable CRENVVAR.EXE to be accessed immediately after the client is RIPLed, before the LAN Requester is started.)

```
NET SHARE RPLFILES=C:\IBMLANRPL /REMARK:"Share for RIPL r/o area" ...  
... /PERMISSIONS:"RWXCDA" /UNLIMITED  
  
NET SHARE WRKFILES=C:\IBMLAN\RPLUSER /REMARK:"Share for RIPL r/w area" ...  
... /PERMISSIONS:"RWXCDA" /UNLIMITED  
  
NET SHARE TLSFILES=D:\CID\EXE\connect /REMARK:"Share for CID" ...  
... /PERMISSIONS:"RWXCDA" /UNLIMITED
```

Figure 56. SRVAUTO.PRO File. With added NET SHARE definition for D:CIDEXEconnect (shared as TLSFILES).

No access profile has to be added for TLSFILES, since it is part of the directory structure covered by the CID alias and you have already created and applied access control profiles for CID.

6. To define the RIPL machines create a master workstation. Use *The Lan Server GUI interface* to define RIPL machines.

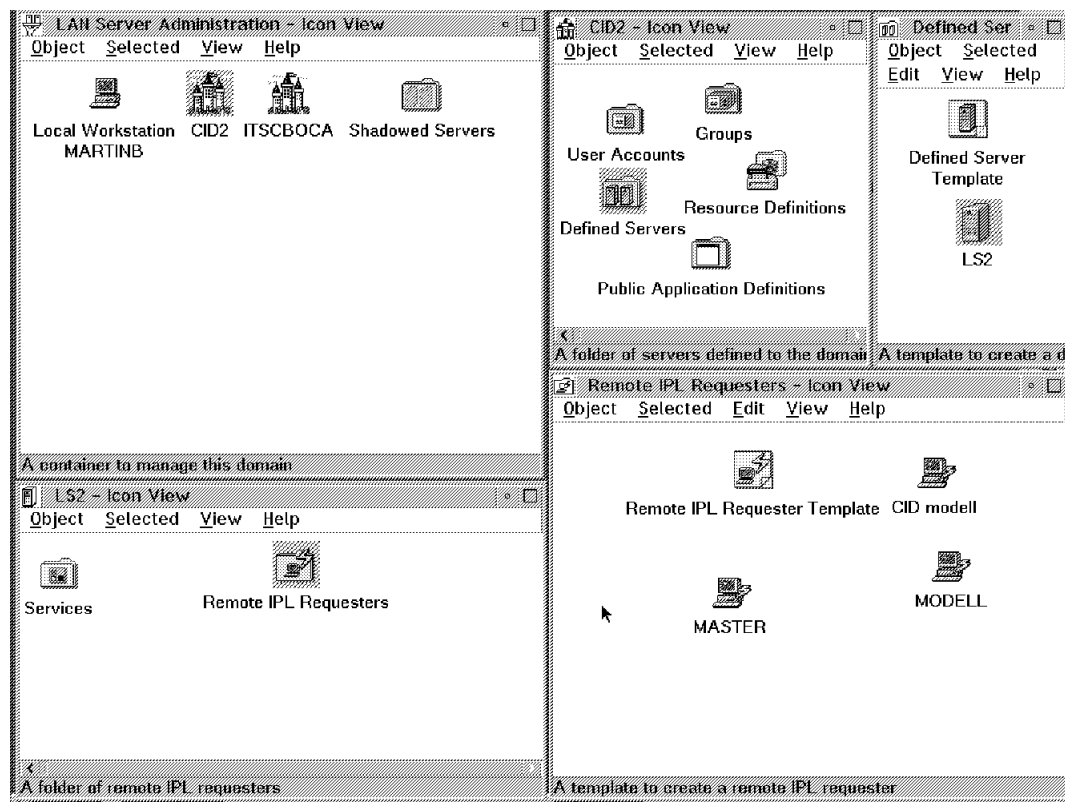


Figure 57. Overview of LAN Server V5.0 GUI.

Open the **Remote IPL Requesters** folder.

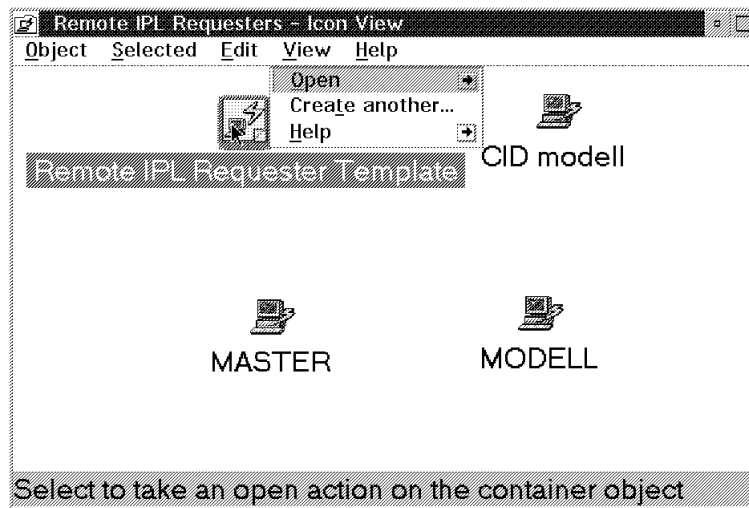


Figure 58. RIPL template menu.

Drag the template to an open area in the folder (or use the context menu *Create another*) which will display the **Remote IPL Create** Notebook.

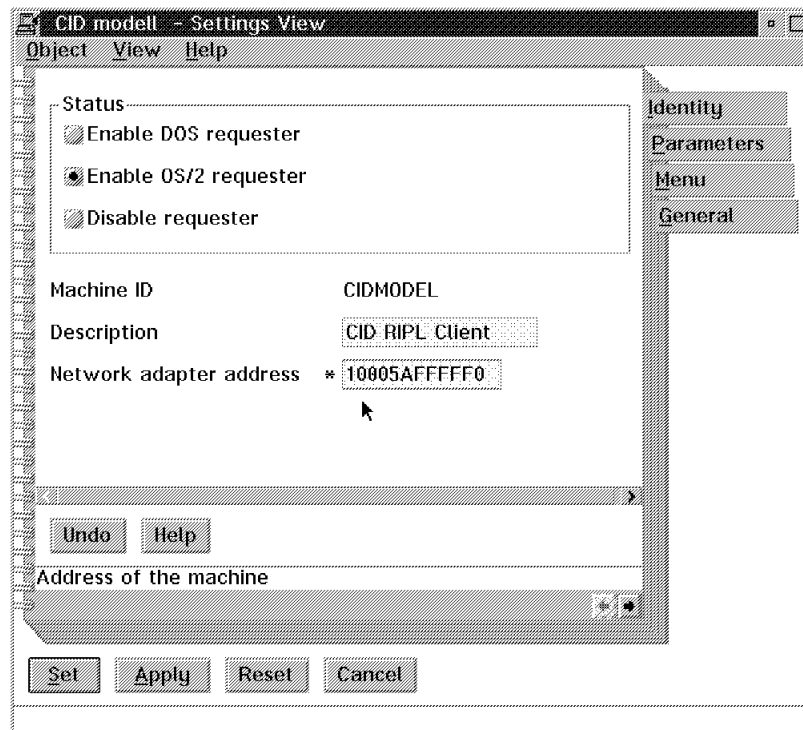


Figure 59. First page of RIPL notebook.

The notebook has four pages:

- Identity
- Parameters
- Menu
- General

a. The **Identity** page

- Click in the Status Field on the Radio button for *Enable OS/2 requester*
- Edit the *machine ID* field and type in CIDMODEL as the name of the model client.
- Under *Description* field type in the machine description,
- Under *Network adapter address* enter the universally administered LAN adapter address.

b. The **Parameters** page

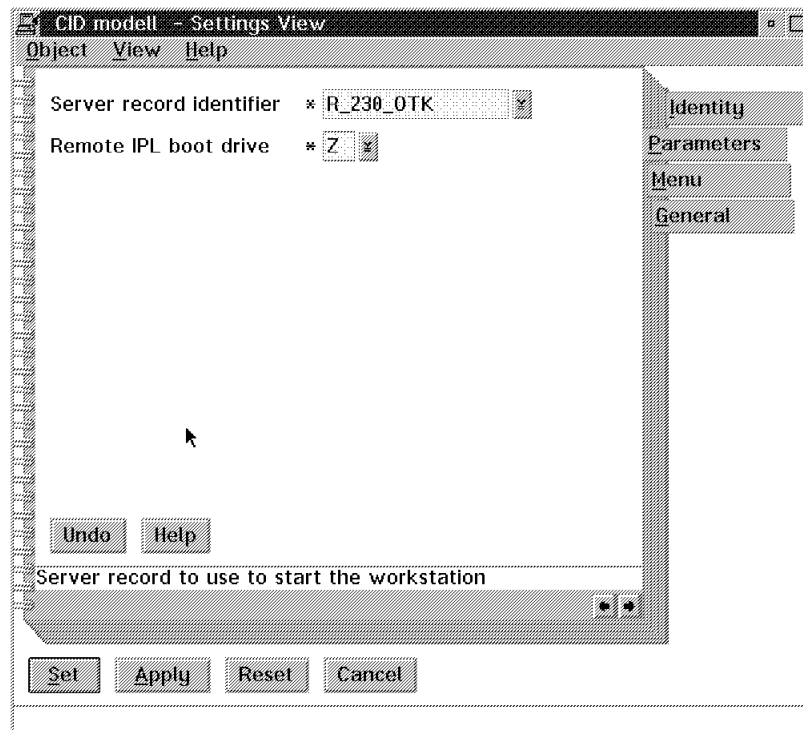


Figure 60. Parameters for RIPL Client.

- Under *server record identifier* field select r_230_OTK,
 - Under *Remote IPL boot drive* field type in Z,
- c. Leave the **Menu** page unchanged
- d. The **General** page

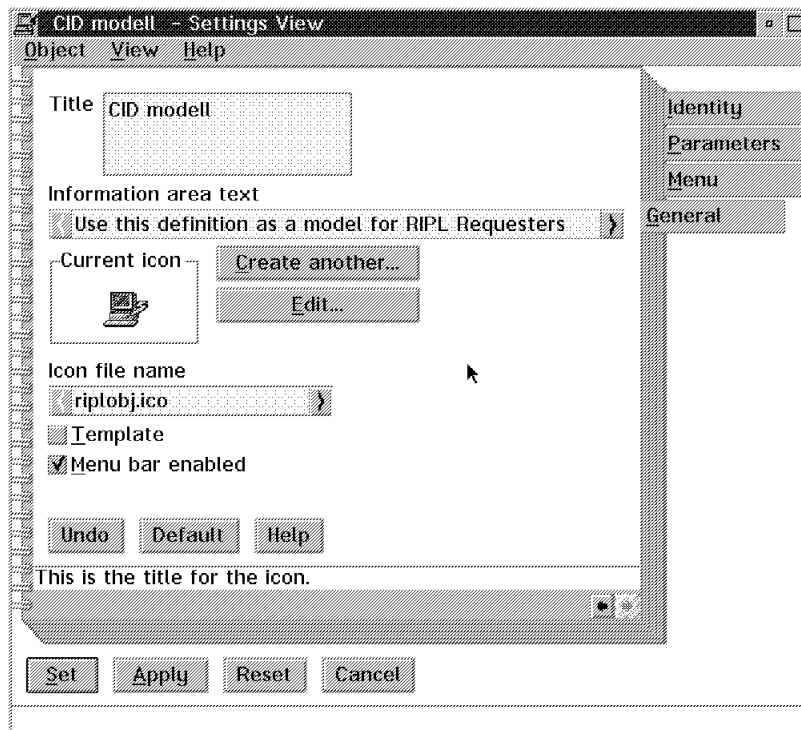


Figure 61. General information about RIPL Client.

- Under *title* field type in the title for the model you create.
- Under *Information area text* field type in information text,
- Leave the rest unchanged.

Click on the **Apply** button to create the model with the parameters you just edited. The machine name entered is automatically added as a *user* in the *User Profile Management* and part of the RPLGROUP.

7. In the IBMLANRPLUSRCIDMODELOS2 path create an INSTALL directory.

This is necessary because during LAPS installation the files LSI.MSG, LSIH.MSG, IBMLANLK.EXE and IBMLANLK.SYS will be copied from the IMGLAPSLANLK to OS2INSTALL (and this must be a directory where the RIPL client has write access).

8. Edit the IBMLANRPLFITSCIDMODEL.FIT file of this master workstation. The following figure shows the additions:

```

LSCIDSRVRPLFILES

; The first line of this file MUST be UNC name

; Per-workstation read-only configuration files.
Z:\CONFIG.SYS          MACHINES\CIDMODEL\CONFIG.30
.
.
; OS/2 Remote Install
Z:\OS2INST             OS2INST
Z:\OS2TOOLS           \\LSCIDSRV\TLSFILES
.
.
; LAN Transport drivers
Z:\IBMCOM              IBMCOM
Z:\PRO.MSG             IBMCOM\PRO.MSG
Z:\IBMCOM\PROTOCOL.INI MACHINES\CIDMODEL\PROTOCOL.INI
Z:\IBMCOM\LANTRAN.LOG  \\LSCIDSRV\WRKFILES\CIDMODEL\IBMCOM\LANTRAN.LOG
; LAPS files copied during LAPS installation
Z:\OS2\INSTALL\LSI.MSG \\LSCIDSRV\WRKFILES\CIDMODEL\OS2\INSTALL
Z:\OS2\INSTALL\LSIH.MSG \\LSCIDSRV\WRKFILES\CIDMODEL\OS2\INSTALL
Z:\OS2\INSTALL\IBMLANLK.EXE \\LSCIDSRV\WRKFILES\CIDMODEL\OS2\INSTALL
Z:\OS2\INSTALL\IBMLANLK.SYS \\LSCIDSRV\WRKFILES\CIDMODEL\OS2\INSTALL
.
.

```

Figure 62. Changes of the CIDMODEL.FIT File. The highlighted entries have to be added.

9. Edit the CONFIG.30 file of the master workstation which is the CONFIG.SYS of the client during the RIPL process. This file is found in the IBMLANRPLMACHINESCIDMODEL subdirectory. The following figure shows the changes in bold text:

```

PROTSHELL=Z:OS2PMSHELL.EXE
.
.
LIBPATH=X:\DLL\connect;X:\IMG\connect\DISK_1;X:\IMG\LCU;.;Z:\OS2\DLL;Z:\IBMLAN\NETLIB;
Z:\MUGLIB\DLL;Z:\OS2\MDOS;Z:\IBMCOM\DLL;Z:\;Z:\OS2\APPS\DLL;Z:\OS2TOOLS;
SET PATH=X:\EXE\connect;X:\IMG\LCU;Z:\OS2;Z:\OS2\SYSTEM;Z:\IBMLAN\NETPROG;Z:\MUGLIB;
Z:\OS2\MDOS\WINOS2;Z:\OS2\INSTALL;Z:\;Z:\OS2\MDOS;Z:\OS2\APPS;Z:\OS2TOOLS;
SET DPATH=X:\EXE\connect;X:\IMG\LCU;Z:\OS2;Z:\OS2\SYSTEM;Z:\IBMLAN\NETPROG;Z:\IBMCOM;
Z:\OS2\MDOS\WINOS2;Z:\OS2\INSTALL;Z:\;Z:\OS2\BITMAP;Z:\OS2\MDOS;Z:\OS2\APPS;Z:\OS2TOOLS;
.
.
REM Use the following statement for SWAPPER.DAT on server:
SWAPPATH=Z:\OS2\SYSTEM 4096 1024
REM Use the following statement for SWAPPER.DAT on workstation:
REM SWAPPATH=C:\ 1024 2048
.

```

Figure 63. CONFIG.30 of the Master Workstation. The highlighted entries need to be added or changed. The additional PATH, DPATH and LIBPATH statements are necessary for the CID process.

The SWAPPATH in this CONFIG.30 should be changed from the default C: to Z:, because the C: drive of the client might be formatted during the OS/2 installation.

10. Create two files STARTRPL.CMD and STARTUP.CMD for the master workstation.

Both files can be found in the CIDIMGSAMPLERIPL subdirectory or in the RIPL subdirectory of the sample code CDROM. Copy them to the IBMLANRPLUSERCIDMODEL subdirectory of the server, which is the home directory of the master workstation.

The STARTRPL is executed after the initial remote IPL process. As described earlier the CRENVVAR.CMD prompts the user for *machine name* (RPCCLIENT in this scenario) and keeps it in ENV_VARS.CMD. A logon to the code server is executed. NET USE for the CID directory structure and for the CIDLOG directory are issued. The SRVREXX.EXE is detached to execute the REXX procedures. Finally, the LCU command file which is the master installation program for the client is invoked.

```
rem Ask for MACHINE/USERID
IF EXIST ENV_VARS.CMD GOTO SETVARS
CRENVVAR /P:"Enter Workstation Name" /V:MACHINE
:SETVARS
CALL ENV_VARS
LOGON %MACHINE% /D:CIDDOM

rem Setup connection to predefined alias

NET USE X: CID
NET USE L: CIDLOG

rem Setup connection to predefined net share (in SRVAUTO.PRO on the server)
rem
rem NET USE X: \\LSCIDSRV\CIDFILES
rem NET USE L: \\LSCIDSRV\LOGFILES

rem Establish REXX functions needed by LCU

DETACH X:\IMG\LCU\SRVREXX

rem Call LCU command file

X:\CLIENT\%MACHINE% %MACHINE% L:%MACHINE%.LOG

rem EXIT
EXIT
```

Figure 64. STARTRPL.CMD File

11. The STARTUP.CMD file contains the call to the STARTRPL.CMD file. This is used to simplify the cleanup of the used files at the end of the CID

process, when only the call to the STARTRPL.CMD file has to be eliminated from the startup procedure of the client.

`CALL STARTRPL.CMD`

Figure 65. STARTUP.CMD File

12. Use NET START RPL to start the REMOTEBOOT service again.

10.3.9 Build Response Files

Response files and the utilities to create them are explained in Chapter 3, "Response Files" on page 47.

When using LAN Server V5.0 RIPL you would at a minimum require proper response files for OS/2, MPTS and LAN Requester V5.0. For OS/2 installations you will probably use a default response file most of the time and not have one response file for each client. For MPTS you will probably have client specific response files (to set the proper LAN address), which will include a "default" response file (where all common keywords are defined). For each client you need to provide a unique LAN Requester V5.0 workstation name (and the domain name), so you will need one response file for each client.

Ensure that you have response files for all products you want to install.

10.3.10 Build Client Installation Control Files

A special LCU REXX command file is called from the client to control the installation process. How the LCU command files are made and work is explained in detail in Chapter 4.4, "LCU Command File" on page 143. Please take some time to read that section carefully, before editing your own LCU command file(s).

10.4 Preparation for RIPL Clients

This section covers step 13 - 16 of the overview.

10.4.1 Preparation of the RIPL Installation Diskette

As pointed out in Chapter 10.1, "Overview of Remote Initial Program Load (RIPL)" on some machines the reference diskette/partition allows the user to change the boot sequence to enable RIPL from the LAN adapter RIPL module. For these machines no diskette is needed.

10.4.1.1 Using MKRDPM Utility

The following procedure is used to create a remote IPL installation diskette. This diskette will be used to simulate the LAN adapter RIPL ROM module, and will be used by the "installers" to initiate installation on the clients.

The LAN adapters supported by MKRDPM are IBM Ethernet, IBM PC Network and IBM Token Ring.

1. Use the DOS FORMAT command (on a DOS booted machine) to create a 1.44MB DOS system diskette.

FORMAT A: /S

The (/S) parameter specifies add DOS system files and create a DOS bootable system diskette.

2. Run the MKRDPM program on the IBM Operating System/2 Local Area Network Server V5.0 to replace DOS system files and create a RIPL bootable diskette.

MKRDPM

3. The RIPL installation diskette is now ready for use.

10.4.1.2 Using RPLENABL Utility

In this case the RIPL installation diskette is the diskette that the "installers" will use to disable the fixed disk of the clients so that they will RIPL from the server and install. The following steps will create the diskette:

1. Create a DOS bootable diskette.
2. Copy the RPLENABL.EXE onto the diskette from the IBMLANRPLDOS directory of the server.
3. Add an AUTOEXEC.BAT to the diskette:

```

@echo off
echo =====
echo Enabling RIPL, please wait.
echo =====
rplenabl
echo =====
echo Remove the Installation diskette
echo from the diskette drive and
echo reboot (Ctrl - Alt - Del)
echo =====
pause > nul
:loop
goto loop

```

Figure 66. RIPL Installation AUTOEXEC.BAT

10.4.2 Test CID RIPL Installation to One Client

To ensure that everything is working use the CIDMODEL RIPL workstation definition to remote IPL and CID install a test machine.

As a side effect the CID clients based on CIDMODEL will RIPL faster! The first time any OS/2 RIPL client is remote IPLed from the LAN Server V5.0 the client's OS/2 desktop is built. Therefore at the first RIPL it takes slightly longer to get the RIPL client up and running than on subsequent RIPLs. So if CIDMODEL is used to RIPL a client once, the desktop is built and it is automatically copied to other RIPL workstation definitions based on CIDMODEL. Therefore, these RIPL clients will RIPL fast even when remote IPLed the first time.

Note.

You can decrease the time for the RIPL process by reducing the environment for the RIPL client. If you set the *OS2_SHELL* to *CMD.EXE* only a fullscreen OS/2 is booted with no Presentation manager. Using this environment only allows you to install products via RIPL that do not need a Presentation manager running.

1. Make the necessary response files for intended CIDMODEL client.
2. Make an LCU command file, CIDMODEL.CMD, to install the selected products.

3. Edit the line in IBMLANRPLRPL.MAP for CIDMODEL and change to the burned-in LAN address for the test client's LAN adapter.

When the client is switched on with RIPL enabled the address can be seen high up on the screen starting with AA and 12 digits. The digits represent the burned-in LAN address.

For example, if the line in RPL.MAP is:

```
10005AFFFFFFD CIDMODEL ~ FITS\CIDMODEL LSCIDSRV Z ~ ~ ~,, ~ R_230_OTK ~ ~ ~
```

you should change the line in RPL.MAP as shown below if the client adapter address is 10005A219C3D:

```
10005A219C3D CIDMODEL ~ FITS\CIDMODEL LSCIDSRV Z ~ ~ ~,, ~ R_230_OTK ~ ~ ~
```

4. RIPL the client and do the test CID installation

Remember that after the first part of the OS/2 installation is made there is a message to "Remove the diskette from drive Z:, and then press <Enter> to reboot". If the client was not RIPLed from the diskette ignore that part of the message otherwise remove the RIPL diskette. Then do a **SHUTDOWN** of the system and when the message appears that it is safe to hit Ctrl+Alt+Del do it in order to reboot the RIPL client. (Pressing Enter just gives you the message again.)

5. When the client is installed and up and running use NET STOP RPL on the LAN Server V5.0 to stop the REMOTEBOOT service again.
6. Change the line in IBMLANRPLRPL.MAP for CIDMODEL back to a dummy address (for example 10005AFFFFFFD).

10.4.3 Create "Remote IPL Workstation" Definitions for Each CID Client

If you are still using LAN Server V3.0 refer to The CID Guide GG24-4295-00 you can find this book in the sample CD under IMG Sub directory. Else, use the LAN Server V5.0 GUI- IBM Lan Services to get to The Remote IPL Requesters Folder, to create remote IPL workstation definitions for the client workstations which will be installed. Refer to 10.3.8, "Code Server Installation." on page 276, 278 and following. Use the **CIDMODEL** RIPL client as a model for all your workstations.

10.5 Running the Code Server

Most tasks on a LAN Server V5.0 can be done through the GUI - IBM Lan Services Icon View or by executing commands at an OS/2 command prompt on the server. Below we will describe the commands as they are done from an OS/2 prompt.

10.5.1 Starting the Code Server

As usual this is done with the NET START SERVER command.

Check in IBMLANIBMLAN.INI file that REMOTEBOOT is defined for SRVSERVICES. Otherwise it is not started automatically whenever the SERVER is started.

If REMOTEBOOT is not started it is not necessary to restart the server; you only have to enter NET START REMOTEBOOT (or NET START RPL).

10.5.2 Stopping the Code Server

If you only want to stop the REMOTEBOOT service issue NET STOP REMOTEBOOT (or NET STOP RPL).

If you want to stop the whole server do NET STOP SERVER. If it has no active sessions it is stopped immediately.

If there are active sessions you will be given the choice of if you want to delete these sessions and force the server to stop.

10.5.3 Display Code Server Status

Active Services

NET START will show you which LAN services are started. NET START or STOP or PAUSE or CONTINUE followed by the service name can be used to manage the service.

Active Sessions

Information about active sessions is shown with the NET SESSION command.

Open files

Information about open files is shown with the NET FILE command.

Logged on users

NET WHO shows the currently logged on users. A RIPLed client does not need to start the LAN requester or logon. For CID installations as described in this book the LAN requester is started on the client and they are logged on in STARTRPL.CMD.

10.6 Customizing the Code Server

To ensure that the CID installation runs as quickly and smoothly as possible check the statistics and error logs to find out if it is necessary to do any tuning of the LAN Server V5.0.

10.6.1 Code Server Security

When the client is RIPLed the security is slightly different than when the LAN requester is active and the user is logged on.

If a client's LAN address is not defined in the IBMLANRPL.MAP it is not RIPLed from the server. Even if it is defined, it has to be an enabled status in order to Remote IPL. You can enable or disable then client in the **Settings** notebook of the client that is similar to the **Create** notebook.

Once it RIPLs, the client's FIT file in IBMLANRPLFITS determines how the client's file requests are resolved. And for the "real files" on the code server the access control profiles are checked before the client gets access.

10.6.2 Working with Authorizations and Client Workstation Names

Each client must be defined as a "RIPL Workstation." See 10.4.3, "Create "Remote IPL Workstation" Definitions for Each CID Client" on page 289. Otherwise a user ID is not defined or added to RPLGROUP and the necessary files are not created.

The ID used for RIPL cannot use a password. Therefore it is not recommended to use the same ID that will be used later for the client when it is installed and wants to connect to some server. If the user connects to another domain for production it can have the same user ID, of course, and be forced to use a password on that domain.

Once it is defined the workstation definition in IBMLANRPL.MAP must be enabled. If the first letter of the workstation type field is R it is active and if it is D it is disabled. If CLIENT1 is enabled the line in RPL.MAP is:

```
10005A219C3D CLIENT1 ~ FITS.CIDMODEL LSCIDSRV Z ~ ~ ,,, ~ R_230_OTK ~ ~ ~
```

and if R_230_OTK is changed to D_230_OTK it is disabled.

Even if it is enabled, the LAN address must be correct. As you see it is easy to disable a client temporarily. Utility

Chapter 11. Manual Setup of LAN CID Utility

This chapter describes the functions of LAN CID Utility (LCU). For the software versions that we used at the time of writing please see Appendix B, "Versions Used in This Book" on page 431.

In our examples we are using OS/2 Warp V3 and MPTS LAN CID Utility.

11.1 IBM Multi-Protocol Transport Services Overview

MPTS provides support for the LAN transport. In addition it also provides the necessary set of utilities for automated installation of OS/2 and other products.

MPTS consists of two different parts:

- LAN Adapter and Protocol Support (LAPS)
- Utilities

These two different parts are physically represented by three diskettes:

1. MPTS diskettes 1 and 2 contains LAN Adapter and Protocol Support. The MPTS diskette 3 contains general LAN transport and CID utilities:

LAPSDDEL.EXE

LAPSDISK.EXE

LAPSRSP.EXE

MPTS.EXE

THINLAPS.EXE

And the unpack utility PKUNZIP2.EXE.

2. MPTS diskette 3 contains CID utilities in two of its subdirectories:

- LCU subdirectory contains LCU.ZIP. Using PKUNZIP2 to unpack LCU.ZIP the following CID utilities will be unpacked (and some related files):

CASAGENT.EXE

CASCKREX.CMD

CASDELET.EXE

CASINSTL.EXE

GETBOOT.CMD
GETOSCID.CMD
GETREXX.CMD
GETFIX.CMD
SRVREXX.EXE

- SRVIFS subdirectory contains SRVIFS.ZIP. Which contains the following CID utilities and related files:

IFSDEL.EXE
SERVICE.EXE
SRVATTCH.EXE
THINIFS.EXE
THINSRV.EXE

In the APPLETS subdirectory there are two ZIP files. CASSETUP.ZIP contains the CASSETUP utility described in Chapter 18, “Automated Setup with CASSETUP” on page 403.

In the MPTSAPLT.ZIP file there is among other things the files for the CASPREP utility discussed in 4.5, “Using LCU CASPREP Utility” on page 169.

For the sample directory there is sample.zip, which contains sample MPTS response files and sample initialization files for a SRVIFS server.

There is a fifth directory with TOOLKIT files, needed if programming for MPTS, but the use of these are beyond the scope of this book.

For a full listing of MPTS diskettes content refer to the MPTS documentation.

11.2 LAN CID Utility Overview

How to use these CID utilities is described on the following pages. A complete quick reference is shown in 17.2, “Quick Reference” on page 400.

The SRVIFS directory contains the files that enable the installation of a simple file server, and the necessary code to install requesters, which can access redirected drives on the server.

The LAN CID Utility is designed to allow an administrator to easily chain together a series of CID installs.

For example, an end user system may require OS/2 Warp Connect, MPTS, LAN Requester V5.0, CM/2 V1.11 and DB2/2 V2.11 Single-User Version to be installed. Though each product is individually enabled for CID, there is the obvious requirement to allow the complete install of all these products to be invoked as a single process instead of several processes. LCU tracks the current state of installation across IPLs and ensures that each CID install program executes in the correct sequence. Once the administrator has defined the desired sequence to install in an LCU command file, the installation process will run to completion without end user involvement at the client workstation. However, an end user must always be at the client workstation to do one of the following:

- Insert the two diskettes created on the server and reboot
- or
- Enable the client workstation to connect to the server and reboot

This is called lightly attended installation; please refer to Chapter 1, "CID History, Concepts and Scenarios" on page 3 for a complete description of the different types of installations in a CID environment.

The LCU files that comprise the software distribution manager are mainly those in the LCU directory.

As shown in the chapters for LAN Server V5.0 RIPL, Novell NetWare and TCP/IP V3.0, it is not necessary to use SRVIFS to achieve the server/requester connection. In those environments for LAN transport the normal requester/server code is used to provide the connections and the remote drives.

LAN CID Utility is used in those environments to provide a software distribution manager.

The following figure shows the relationship between the client workstation and the code server.

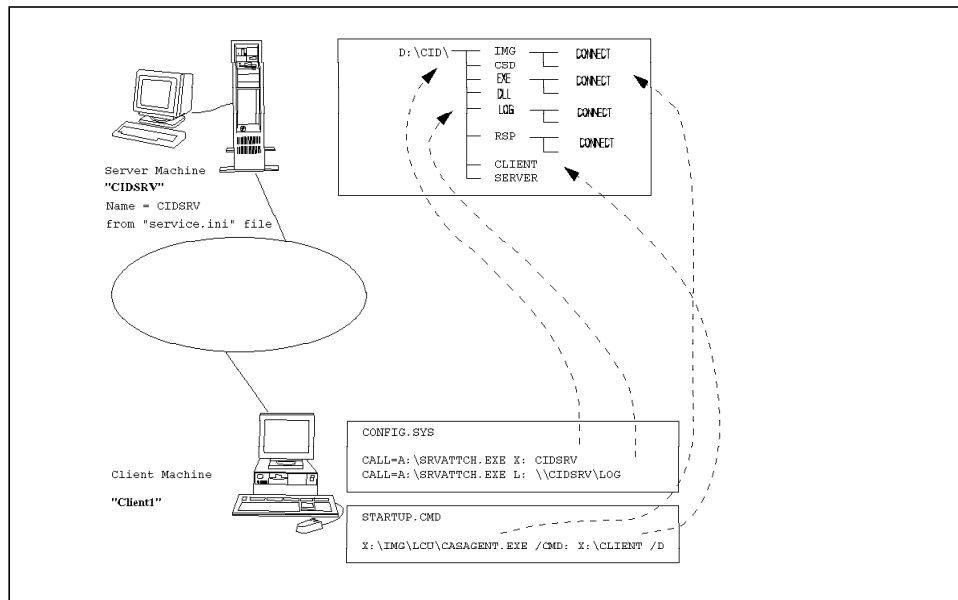


Figure 67. LAN CID Utility Environment Using SRVIFS

11.3 Scenario

In the rest of this chapter we will walk you through the manual installation of a code server using LAN CID Utility to install OS/2 Warp Connect, MPTS, CM/2 V1.11, PC/3270 for OS/2 V4.1, CM Server V4.0, DB2/2 V2.11 Single-User Version and LAN Server V5.0.

Overview of installation steps for SRVIFS based LCU server:

1. Install OS/2 Warp Connect and MPTS.
2. Create a code server directory structure.
3. Load OS/2 CID Utilities.
4. Load the product diskette images using the product dependent procedures.
5. Load EXE- and DLL-files that are used to support the clients during installation.
6. Install the LAN CID Utility.
7. Build product dependent response files.
8. Build LCU command files.

9. Create client boot diskettes for diskette-initiated installations.
10. Start the LCU code server.
11. The client boots from diskettes and installs.

11.4 Manual Installation

11.4.1 Basic Installation of Code Server

See Table 11 on page 266 for the required software. The only software that must be installed on the code server are OS/2 and MPTS.

11.4.2 Creating the CID Directory Structure

The recommended common CID directory structure to be used with LCU is described in Chapter 2, "Recommended CID Directory Structure" on page 39. This common directory structure has been developed for the CID process to ensure compatibility/migration between LAN CID Utility, and NetView Distribution Manager/2. LCU, by itself, does not require any fixed directory structure. However, we recommend the use of the common CID directory structure to avoid any compatibility problems with follow-on products that will be shipped in the future.

Ensure that the disk you want to use has enough free space to hold the desired product images and has additional space available for response, LCU command and log files. See Table 10 on page 264 for a listing of the space needed by the product images.

Use **MKDIR** or **MD** to create the directory structure.

When you follow the examples later in the text please remember to use the correct drive letter. The examples assume that the disk with the CID directory structure is D:.

11.4.3 Loading OS/2 CID Utilities to Code Server

Please see Chapter 15, "OS/2 CID Utilities" on page 373 on how to load OS/2 CID Utilities to the LCU code server.

11.4.4 Copy Files from the Sample Code CDROM to Code Server

You can manage without using anything from the sample code CDROM, but then you have to do **every** step manually. As you might have read in Part 2 of this book there are some nice utilities on the sample code CDROM.

Assuming that the sample code CDROM is accessed as E: enter the following commands:

```
XCOPY E:\utility D:\cidexeconnect
XCOPY E:\sample D:\cid\img\sample
XCOPY E:\srvifs\cidshr.ini D:\cid\img\srvifs
COPY E:\srvifs\connect.cmd D:\cid\client
```

If \cid\server does not yet exist, the XCOPY of CIDSRV.INI will ask you if this is intended to be to a directory and you should reply with a yes.

11.4.5 Loading Diskette Images

Please see Chapter 16, "Loading Product Images to Code Server" on page 379 on how to load the product diskette images to the LCU code server.

The minimum requirements when using an LCU code server are that you load:

- OS/2 Warp Connect. See 16.1.1, "Loading OS/2 Diskette Images with SEIMAGE" on page 379.
- IBM Multi-Protocol Transport Services. See 16.1.2, "Loading LAN Transport System Diskette Image(s) with LAPSDISK" on page 382.
- LCU files. See 16.1.8, "Loading LAN CID Utility Files" on page 394.
- SRVIFS files. See 16.1.9, "Loading SRVIFS Files" on page 394.

11.4.6 Copy REXX to Code Server

GETREXX helps you copy the REXX support necessary for the clients to be able to run their command files. See 17.1.2, "GETREXX" on page 398.

11.4.7 Copy SETBOOT.EXE and XCOPY.EXE to Code Server

GETBOOT helps you copy SETBOOT.EXE and XCOPY.EXE, which are necessary for the clients to be able to run their command files. See 17.1.1, "GETBOOT" on page 397.

11.4.8 Code Server Installation (THINSRV)

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 startup.

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 target's CONFIG.SYS file. THINSRV will also add a START statement to the target's STARTUP.CMD file.

THINSRV Syntax

THINSRV /R: /T: /S: /TU: /U: /L1:

/R: **Fully qualified path and name of a code server response file**

The format and the contents of the code server response file are documented in Appendix L, "The SERVICE.INI File Keywords" on page 605.

This response file is copied to the target directory and is renamed with the file extension INI.

THINSRV will ensure that both the value of the path statement and the value of the path in the alias statement exist.

THINSRV will terminate if required configuration parameters are missing from the **INI** file.

If this parameter is not furnished THINSRV will terminate.

Note on SERVICE.INI

There is a sample SERVICE.INI file shipped on the MPTS Utility diskette in the SAMPLE directory. From SAMPLESAMPLE.ZIP on MPTS diskette 3, PKUNZIP2 can be used to unpack a sample SERVICE.INI and a SERVICE.LST.

The administrator can create tailored versions of SERVICE.INI. These are either based on the sample SERVICE.INI file **or** on the CIDSRV.INI from the sample code distributed with this document.

/T: Fully qualified target path

This parameter is optional.

If the fully qualified path is omitted, it will default to the current boot drive of the system.

THINSRV will attempt to create the target subdirectory if it does not exist. If the target is invalid or unable to create the target subdirectory, THINSRV will terminate.

/S: Fully qualified source path

On an installed code server the SRVIFS source files are usually in the IMGSRVIFS directory.

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

/TU: Fully qualified path to CONFIG.SYS

Value supplied is the fully qualified path of the CONFIG.SYS that will be updated with LCU CID code server statements. If /TU: parameter is omitted, the value for /T: parameter will be used as the default.

THINSRV will create a STARTUP.CMD on this drive, if it does not exist, and adds a statement to start the SRVIFS server.

THINSRV will terminate if a CONFIG.SYS does not exist in the determined location.

/U: Authlist file source

This parameter is optional.

Value supplied is the name of the authentication list (authlist) file granting access to the SRVIFS code server.

The authlist file will be copied from the source pointed to by this parameter to the target as specified by the AuthList keyword

value in the response (INI) file. AuthList has to be defined for a copy to take place.

If the target subdirectory does not exist, THINSRV will create the subdirectory.

THINSRV will terminate if the source file cannot be located.

A default authlist file will be located in the same path as the response file and the file name indicated in the response (INI) file.

MPTS provides a sample authorization list SERVICE.LST.

/L1: Log file name

This parameter is optional.

Value supplied is the fully qualified path and file name of the log file.

THINSRV Example

```
D:\cid\img\srvinfos\THINSRV
/R:D:\cid\img\sample\srvinfos\cidsrv.ini
/T:D:\cid\server
/S:A: /TU:C:\
/U:D:\cid\img\srvinfos\service.lst
/L1:D:\cid\log\srvinfos\thinsrv.log
```

The command above copies the necessary code server files into the D:\CIDSERVER directory. It will also create a valid CIDSERV.INI file based on the response file name specified in the /R parameter. THINSRV validates the content of the response file before creating the actual CIDSERV.INI file used by SERVICE.EXE.

Note on THINSRV

It is NOT necessary to reboot the code server workstation to be able to activate the LCU code server. The administrator can execute the STARTUP.CMD or the SERVICE command from the command line. For more information see 11.6.1, "Starting the Code Server" on page 311.

11.4.9 Build Response Files

Response files and the utilities to create them are explained in Chapter 3, “Response Files” on page 47.

When using LCU you would at a minimum require proper response files for OS/2 and MPTS. For OS/2 installations you will probably use a default response file most of the time and not have one response file for each client. For MPTS you will probably have client specific response files (to set the proper LAN address), which will include a default response file (where all common keywords are defined).

Ensure that you have response files for all products you want to install.

11.4.10 Build Client Installation Control Files

A special LCU REXX command file is called from the LCU agent running on the client to control the installation process. How the LCU command files are made and work are explained in detail in 4.4, “LCU Command File” on page 143. Please take some time to read that section carefully, before creating your own LCU command file(s).

11.5 Preparation of Client Workstations

This section describes the creation of the LCU redirector boot diskettes. The LCU redirector boot diskettes are prepared by the code server administrator on the code server workstation.

11.5.1 Running SEDISK

To create the bootable OS/2 diskettes use 15.1.2, “SEDISK” on page 377.

11.5.2 Adding LAN Transport System to Client Diskettes (THINLAPS)

In order to transfer NetBIOS and the network drivers onto the LTS diskette, the code server administrator uses a utility called THINLAPS. For a detailed description refer to 4.1.2.3, “THINLAPS” on page 92.

THINLAPS will install a seed LAN transport system on the LTS diskette and update the CONFIG.SYS accordingly.

— THINLAPS example for client with IBM Token-Ring adapter —

```
D:\cid\img\laps\THINLAPS D:\cid\img\laps A:\ ibmtok.nif
```


A PROTOCOL.INI is created on the target LTS diskette based on a valid Network Information File (.NIF). The name of the .NIF file is supplied as a parameter. See Appendix E, "LAN Network Adapters" on page 489 for a mapping of the LAN transport network adapter device drivers and the associated .NIF file names. The following figure shows the PROTOCOL.INI created on the LTS diskette based on the **IBMTOK.NIF** parameter.

```
[protman]
  drivename = protman$

[netbeui_nif]
  drivename = netbeui$ bindings = mac

[mac]
  drivename = 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
```

Figure 68. PROTOCOL.INI File of the LTS Diskette after MPTS THINLAPS

The default PROTOCOL.INI file created by THINLAPS is only valid for Micro Channel adapters; do not forget to remove the semicolon and append the correct value of the shared RAM address space if you plan to use the boot diskettes for installation on ISA-bus machines.

The following figure shows the LTS 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,vtb1850.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.flr
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 69. CONFIG.SYS File of the LTS Diskette after MPTS THINLAPS

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. For more information on how to do this see Appendix E, "LAN Network Adapters" on page 489.

Note

For example, changes necessary for NetBIOS support on Ethernet:

If there are two different types of network adapters installed in the client workstation the token-ring needs to be set to run secondary and the Ethernet primary.

11.5.2.1 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 LAPS diskette image as described in 16.1.2, "Loading LAN Transport System Diskette Image(s) with LAPSDISK" on page 382. Therefore it is necessary to add them by following the instructions in Appendix E, "LAN Network Adapters" on page 489.

11.5.3 Install LCU Redirector (THINIFS)

SRVIFS client/redirector In order to transfer the SRVIFS redirector code to the LTS diskette, the code server administrator uses a utility called THINIFS. For a detailed description of THINIFS refer to 4.1.3.1, "THINIFS" on page 94. THINIFS will install the SRVIFS redirector files to the LTS diskette and update the CONFIG.SYS accordingly.

First THINIFS example

```
D:\cid\img\srvifs\THINIFS /S:D:\cid\img\srvifs /T:A:
/SRV:CIDSRV /REQ:CLIENT1 /D:X:
```

The name function, /REQ on the IFS command in the CONFIG.SYS works in three ways:

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

MPTS CASAGENT 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.)

In our example CLIENT1 is used as the SRVIFS client name.

The first invocation of THINIFS will add a SRVATTCH statement to the LCU redirector's CONFIG.SYS file. This statement points to the default path D:\CID on the code server and will be accessed by the LCU redirector as drive X:.

See SERVICE.INI parameters in Figure 121 on page 611 for the definition of the default path. Note that D:\CID does not permit writing, as it is shared as "read only". See also Figure 70 on page 307 for the SRVATTCH as drive X: definition.

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.

Second THINIFS example

```
D:\cid\img\srvifs\THINIFS /S:D:\cid\img\srvifs /T:A:  
/SRV:\\cidsrv\log /REQ:CLIENT1 /D:L:
```

The second invocation of THINIFS will add a second SRVATTCH statement to the LCU redirector's CONFIG.SYS file. This statement points to the LOG alias CIDSRVLOG on the code server and will be accessed by the LCU redirector as drive L:.

Figure 121 on page 611 for the definition of the LOG alias.

Note on THINIFS

The second invocation of THINIFS will move the first SRVATTCH statement **CALL=A:SRVATTCH.EXE X: CIDSRV** before the DEVICE and IFS statements. This is acceptable since IFS and DEVICE statements will be processed before the CALL statement.

Figure 70 on page 307 shows the CONFIG.SYS on the LTS diskette file after the second THINIFS:

```
.  
.   
.   
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  
DEVICE=A:\SRVIFS.SYS  
IFS=A:\SRVIFSC.IFS CLIENT1  
CALL=A:\SRVATTCH.EXE L:  \\CIDSRV\LOG
```

Figure 70. Last Part of CONFIG.SYS File of the LTS Diskette after Second THINIFS

11.5.4 Install LCU client (CASINSTL)

CASINSTL will create STARTUP.CMD and update the CONFIG.SYS on the LTS diskette.

CASINSTL example

```
D:\cid\img\lcu\CASINSTL  
  /TU:A: /CMD:X:\client  
  /PL:X:\dll\connect;X:\img\lcu  
  /PA:X:\img\lcu /D
```

For a detailed description of CASINSTL refer to 4.1.3.3, "CASINSTL" on page 101. If MPTS CASINSTL is used some additional parameters are supported by CASINSTL.

Figure 71 on page 308 shows the LTS diskette CONFIG.SYS file after MPTS CASINSTL:

```
buffers=32
iop1=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\CONNECT;X:\IMG\LCU;
ifs=hpfs.ifs /c:64
pauseonerror=no
codepage=850
devinfo=kbd,us,keyboard.dcp
devinfo=scr,ega,vtb1850.dcp
device=\dos.sys
set path=.\;\os2;\os2\system;\os2\install;A;;
set dpath=.\;\os2;\os2\system;\os2\install;A;;X:\DLL\CONNECT;X:\IMG\LCU;
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
device=\testcfg.sys
basedev=xdfloppy.fl1t
device=\refpart.sys
```

Figure 71 (Part 1 of 2). CONFIG.SYS File of the LTS Diskette after CASINSTL

```

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 71 (Part 2 of 2). CONFIG.SYS File of the LTS Diskette after CASINSTL

```

X:\IMG\LCU\CASAGENT.EXE /CMD:X:\CLIENT /D
CMD
EXIT

```

Figure 72. STARTUP.CMD File of the LTS Diskette Created by CASINSTL

CASINSTL does not copy files to the LTS diskette. CONFIG.SYS executes **SRVREXX** located in the code server's executable directory. SRVREXX is a utility loading REXX support from the code server into client workstation memory in order to run REXX LCU command files.

STARTUP.CMD executes the LCU agent **CASAGENT** located in the code server's executable directory. CASAGENT will search in X:\CLIENT for an LCU command file having the same name as the <CLIENT_NAME> specified in the IFS=A:\SRVIFSC.IFS client_name statement in CONFIG.SYS, or if *P was used instead of <CLIENT_NAME>, the name the user is prompted for. The /D parameter tells CASAGENT to search for an LCU command file named DEFAULT.CMD if <CLIENT_NAME>.CMD does not exist. If DEFAULT.CMD does not exist CASAGENT aborts and exits STARTUP.CMD. MPTS CASINSTL supports the /D:<commandfile> parameter. If this parameter is specified, CASAGENT executes the given LCU command file. If it does not exist, CASAGENT aborts and exits.

MPTS CASINSTL supports a new parameter /PD: which will point to a directory on the code server, which contains a copy of the files from boot diskette 1.

During the installation the user at the client machine will be prompted to first change from the LTS installation diskette to diskette 1. By using a redirected diskette 1, the prompt displays just after the CASAGENT program starts, otherwise, the prompt displays before the first reboot is to occur. After the last diskette is removed, the installation continues without further interaction.

11.5.4.1 Copy Boot Diskette 1 to the Code Server

If CASINSTL is used with /PD LTS diskette 1 needs to be copied to a directory on the code server. For OS/2 Warp Connect this would be D:CIDDISK1connect. Assuming the LTS diskette 1 is in drive A: the command to copy the files would be:

```
COPY A:*. *D:CIDDISK1connect.
```

The invocation of CASINSTL would be as shown in the example below.

CASINSTL with /PD Example

```
D:\cid\img\lcu\CASINSTL
/TU:A: /CMD:X:\client
/PL:X:\dll\connect;X:\img\lcu
/PA:X:\img\lcu
/PD:X:\disk1\connect /D
```

11.6 Running the Code Server (SERVICE.EXE)

The LCU SERVICE.EXE is a simple LAN server, which can share drives and directories with aliases and provide security functions. The client redirector is activated through the SRVIFS statements in the client's CONFIG.SYS. The client's redirected drives are accessed through the SRVATTCH statements; see Figure 70 on page 307.

This section will describe how to start, stop, query the status of the code server and refresh the authorization list.

SERVICE.EXE Syntax

```
SERVICE [/INI [/ST | /QU | /AU | /F]]
```


11.6.1 Starting the Code Server

In order to start the code server, enter:

```
SERVICE /INI=NAME
```

NAME specifies the name of the INI file. *NAME* is a 1 to 8 alphanumeric character string.

Note: The full name of the file is *NAME.INI*, but the *.INI* is always omitted.

If the */INI* parameter is not specified, the default file *SERVICE.INI* will be used.

Note: The name *SERVICE.INI* is used in examples in the text in many places, but as seen above your INI file could be named differently.

Multiple servers can be started on the same machine using multiple *.INI* files. For example:

```
SERVICE /INI=SERVER1  
SERVICE /INI=SERVER2
```

Here two INI files are required. One is called *SERVER1.INI* and another is called *SERVER2.INI*.

11.6.2 Stopping the Code Server

In order to stop the code server, enter from an OS/2 command prompt:

```
SERVICE /INI=NAME /Q
```

/Q (QUIT) will ask the server to stop taking new requests and shut down when the last client disconnects.

11.6.3 Display Code Server Status

In order to display the code server status, enter from an OS/2 command prompt:

```
SERVICE /INI=NAME /ST
```

Output from the display includes the values from the *NAME.INI* file, the number of active clients, client names, number of open files and the current directory in process (being installed on the client workstation).

11.6.4 Refresh Authorization List of Code Server

In order to refresh the authorization list, enter from an OS/2 command prompt:

```
SERVICE /INI=NAME /AU
```

The code server administrator can update the authorization list file and issue this command in order to control code server access without having to stop and restart the code server.

11.6.5 Forcing Code Server to Stop

In order to force the code server to stop even when clients are connected, enter from an OS/2 command prompt:

```
SERVICE /INI=NAME /F
```

/F (FORCE) will ask the server to stop taking new requests and shut down immediately even if clients are connected.

11.7 Customizing the Code Server

11.7.1 Code Server Security

The common CID directory structure permits limited security for the code server. This can be achieved with the LCU SRVATTCH command if logs are kept separate from the other data. Log files are the only files in the directory structure that are "read/write"; therefore, the directory structure puts the log files into a separate LOG subdirectory. By doing this the administrator can set up an additional SRVATTCH drive alias for the log files that is "read/write". Any other aliases for product images, response files, executables, and LCU command files are "read only". For examples see L.2, "The Use of Redirected Drives with Aliases" on page 612.

In order for LCU to be able to log prior to the LCU command file being called, LCU must have access to the "read/write" subdirectory that will contain the log files.

11.7.1.1 Working with Authorization Lists and Client Workstation Names

AuthList keyword of the SERVICE.INI file and the name parameter of the IFS command in the client's CONFIG.SYS file go hand in hand. Together they control access between client and code server. This ensures that only authorized clients have access to particular code server(s) and that the clients use correct LCU command files and response files.

The *AuthList* feature supports the use of a name to identify a client. If the name is **not** in the list, access to the server is denied.

By adding the token-ring adapter address to each client name in the authorization list file, the administrator ties a client name to a specific workstation. This can be used to control workstation access to the server and eliminates the possibility of a workstation calling the wrong client response file.

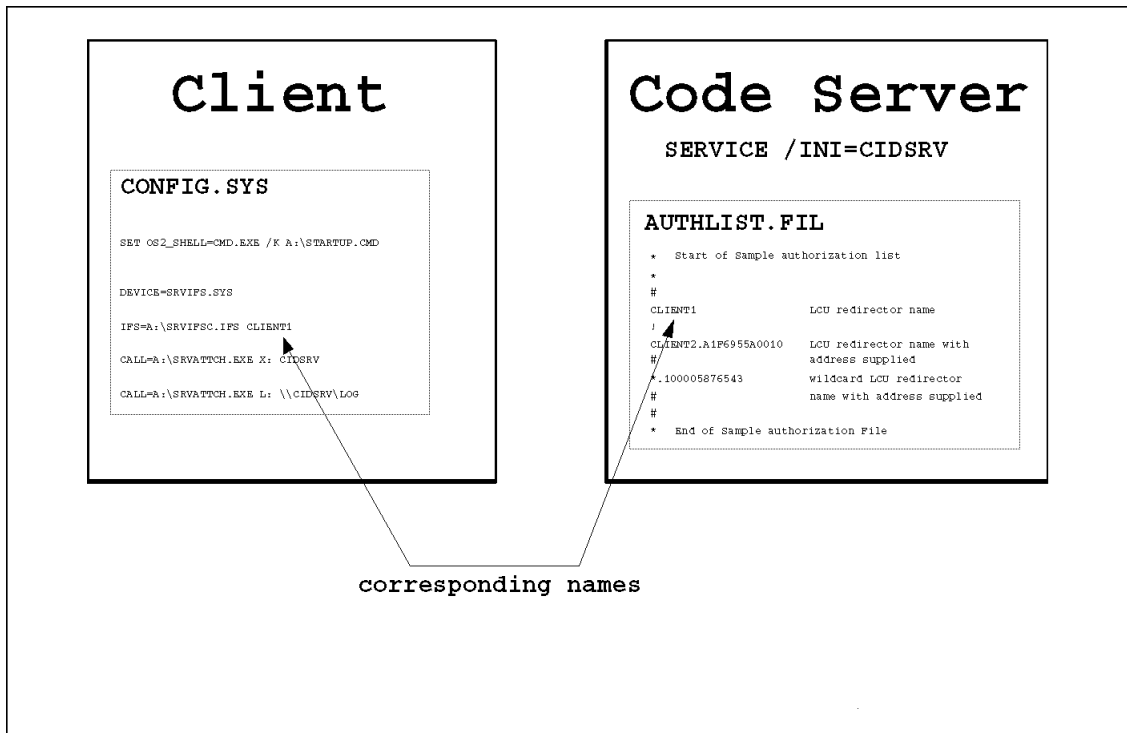


Figure 73. Relation between AuthList and IFS Statement

The client name parameter alone provides security as explained in the following section. The combined use of *AuthList* keyword and client name

provides security checking down to the level of the universally administered address of the token-ring adapter.

The following table gives a brief overview on how these two functions work with each other, showing which combinations are valid and which are not, and the level of security given with a specific selection.

<i>Table 12. Security by Use of AuthList Keyword and Client Workstation Name</i>		
	With Names	Without Names
AuthList activated	high	invalid combination
AuthList not activated	low	none

11.7.2 Customizing Client Diskettes

11.7.2.1 Individual LAN Transport Services Diskettes

The tailoring of individual LTS diskettes means as many LTS diskettes need to be created as there are workstations on the LAN. With a large number of installations and workstations this could prove impractical.

The administrator's work consists of creating the authorization list(s), the appropriate response files and appropriate LCU command files. The administrator can decide to distribute as many LTS diskettes as there are client workstation names in the authorization file. Each one of which has the actual workstation name written out on the IFS statement in the CONFIG.SYS. It is also a good idea to write it on the external diskette label.

11.7.2.2 Common LAN Transport Services Diskettes

When the user is prompted for the client name it is much easier to administrate as only **one** type of LTS diskette exists, which could be copied as many times as needed. By having the user type in the workstation name the individuality would come when the access to the LCU command file is done.

In this case the administrator's work would consist of creating the authorization lists and only one type of LTS diskette. Together with each authorization list the administrator has to build the individual LCU command file and response files.

In this case only universally administered token-ring adapter addresses could be used in the authorization list, since the PROTOCOL.INI on the LTS

diskette should not set any address when it might be used by several users from several machines at the same time.

Note: Of course there is a need for one set of LTS diskettes for each type of LAN adapter used for installations.

11.7.3 Customizing the SERVICE.INI file

It is necessary to set SERVICE.INI. file parameters to define:

- The server name
- The adapter(s) supported
- The maximum number of concurrently active clients
- The maximum number of concurrently open files
- The number of threads used to support client requests
- Which clients have access to the server (optional)
- Directory aliases and access type
- Logging requirements

Normally, the SERVICE.INI file will be in the same directory, D:\CIDSERVER, as the **SERVICE.EXE** file.

The INI file keyword definitions and a sample SERVICE.INI file are documented in Appendix L, "The SERVICE.INI File Keywords" on page 605.

Chapter 12. Manual Setup of Novell NetWare

This chapter will describe the setup of a Novell NetWare server to install OS/2 V2.x and related products for a remote install using the CID installation method. For the software versions that we used at the time of writing please see Appendix B, "Versions Used in This Book" on page 431.

Note

The following chapter is only valid for the Novell NetWare server Version 3.11 /3.12. We did not test an OS/2 Warp V3 or OS/2 Warp Connect installation with this scenario although it may be possible to use this environment to install OS/2 Warp V3. With the newer Novell NetWare server version 4.10 we recommend to use the IBM NetView Distribution Manager for NetWare Version 1.1 to setup a CID-scenario. For detailed information about Software Distribution and installation see : *Software Distribution Using Netview Distribution Manager for NetWare* GG24-4416.

12.1 Overview

The CID installation method uses the concept of redirected drives to gain access to the images that are accessed on a server and a response file that contains all necessary configuration information as input to the installation. There is little or no user intervention required once the process has started. The disk images and the response files will be read from the server. This requires a special setup of the server:

- Creation of the recommended CID directory structure on the server
- Copying of the diskette images onto the server
- Creation of LCU command and response files for the clients
- Creation of the LAN transport system diskettes

In this chapter we will show how to use a Novell NetWare server as a CID code server that runs the LAN CID Utility as a tool for the remote install of CID-enabled products. For a detailed description on how the NTS/2 LAN CID Utility works, please refer to 4.4, "LCU Command File" on page 143.

12.2 Scenario

In this chapter we will walk you through the manual installation of a code server using a Novell NetWare server to install OS/2 V2.11, LAPS, and NetWare Workstation for OS/2 V2.01. The term NetWare requester is used to shortcut the official product name of *NetWare Workstation for OS/2 V2.01*.

It is assumed that the reader following the procedure in this chapter is familiar with the Novell NetWare administrative terms and commands. If this is not the case, it is recommended that a copy of the Novell NetWare administrator's manual and/or quick reference guide be available. In this scenario there will be no detailed description of the administrative tasks concerning the definition of users, assigning of rights and access permissions.

It is assumed that Novell NetWare Server V4.0 is installed and running. It is also assumed that the following steps are executed from a workstation running OS/2 and NetWare requester that is connected to the NetWare server performing the installations. You need to be logged in to the NetWare server as a user with supervisory permissions. To perform the NETADMIN function it is necessary with the current versions of the NetWare requester to start a workstation with DOS and NetWare requester. You will need this function to create users and give these clients the necessary permissions to get access to the CID directory tree.

Overview of Installation Steps for Novell NetWare:

1. Install OS/2 V2.11 and NetWare Workstation for OS/2 V2.01 and get access to the NetWare server.
2. Create a code server directory structure.
3. Load OS/2 CID Utilities.
4. Load the product diskette images using the product dependent procedures.
5. Load EXE and DLL files that are used to support the clients during installation.
6. Install the LAN CID Utility.
7. Build product dependent response files.
8. Build LCU command files.
9. Create client boot diskettes for diskette-initiated workstations.
10. The client boots from diskettes and installs.

Note on “NetWare Workstation for OS/2 V2.1”

We tried to run the scenario described here with NetWare Workstation for OS/2 V2.1 but we were not successful. The client boot diskettes created with V.2.10 did not connect to the NetWare server. We assume that the NetWare Requester V2.10 needs PM to be available to execute. If you want to run this scenario with V2.10 you should create client boot diskettes from V2.01. For the install performed as soon as the client is connected to the server you can use the files of V2.10.

12.3 Manual Installation

12.3.1 Basic Installation of NetWare Server and NetWare Requester

Please refer to the product manuals for information about installing a Novell NetWare server.

A workstation running OS/2 and NetWare Workstation for OS/2 V2.01 is needed. The user ID logged in to the NetWare server from this workstation should have the required permissions to create and modify directories and files. We used the predefined ADMIN user ID to create the CID structure. A drive letter should be mapped that is chosen to hold the directory structure.

The server name used in this scenario was NETWARE40. When following the examples later in the text please remember to use the correct drive letter. The examples assume that the disk with the CID directory structure was mapped as drive **D:**. See Table 11 on page 266 for the required software. If you already have a NetWare server working as a code server for OS/2 V2.0 you can still use it as long as you take care to use the version dependent utilities as noted in the steps below. See also Chapter 19, “Migration and How to Add New Products” on page 405.

12.3.2 Creating the CID Directory Structure

The recommended common CID directory structure to be used with LCU is described in Chapter 2, “Recommended CID Directory Structure” on page 39. This common directory structure has been developed for the CID process to ensure compatibility/migration between LAN CID Utility, and IBM NetView Distribution Manager/2 Version 2.0. LCU, by itself, does not require any fixed directory structure. However, we recommend the use of the CID common directory structure to avoid any compatibility problems with follow-on products that will be shipped in the future.

Ensure that the disk you want to use has enough free space to hold the desired product images and has additional space available for response, LCU command and log files. See Table 10 on page 264.

Use **MKDIR** or **MD** to create the directory structure.

12.3.3 Loading OS/2 CID Utilities to Code Server

Please see Chapter 15, “OS/2 CID Utilities” on page 373 on how to load OS/2 CID Utilities to the code server.

12.3.4 Loading Diskette Images

Please see Chapter 16, “Loading Product Images to Code Server” on page 379 on how to load the product diskette images to the code server. You will need at least the following diskette images to perform a successful install:

- OS/2 V2.11. See 16.1.1, “Loading OS/2 Diskette Images with SEIMAGE” on page 379.
- LAN Adapter and Protocol Support. See 16.1.2, “Loading LAN Transport System Diskette Image(s) with LAPSDISK” on page 382.
- Novell NetWare Requester. See 16.1.10, “Loading NetWare Requester Files” on page 395.

Finally, you should copy the contents of the sample code CDROM to the D:\CIDIMGSAMPLE subdirectory.

```
XCOPY A:\NETWARE D:\CIDIMGSAMPLE\NETWARE /V/S/E
XCOPY A:\UTILITY D:\CID\IMG\SAMPLE /V/S/E
```

12.3.5 Copy REXX to Code Server

GETREXX helps you copy the REXX support necessary for the clients to be able to run their command files. See 17.1.2, “GETREXX” on page 398.

12.3.6 Copy SETBOOT.EXE to Code Server

GETBOOT helps you copy SETBOOT.EXE and XCOPY.EXE, which are necessary for the clients to be able to run their command files. See 17.1.1, “GETBOOT” on page 397.

12.3.7 Code Server Installation

As we use the transport mechanism of Novell NetWare to connect the code server to the clients there is no additional server installation necessary. Clients, however, still have to be defined by the administrator, mappings have to be prepared for the CID directory structure and permissions have to be given to the clients:

- Read and FileScan permissions for the CID directory
- Create, Write, FileScan, Read permissions for the CIDLOG subdirectory

It is also useful to define LOGIN scripts for the clients to provide access to the NetWare utilities that are needed to map additional drives.

— LOGIN Script —

```
MAP L:=SYS:PUBLIC:
```

12.3.8 Build Response Files

Response files and the utilities to create them are explained in Chapter 3, "Response Files" on page 47.

When using LCU you would at a minimum require proper response files for OS/2 and LAPS. As the NetWare requester is not yet CID-enabled there is no response file needed for the install, though there might be a need to supply client-specific files like NET.CFG. If you have a need to supply these files, this can be added to the procedures described in 4.2.1, "Installation of Novell NetWare Workstation for OS/2 V2.01" on page 128. For OS/2 installations you will probably use a default response file most of the time and not have one response file for each client.

There is a sample LAPS response file named LAPSRSP.RSP provided in the NetWare directory of the sample code CDROM. This sample LAPS response file reflects the special setup needed in the NetWare environment with the use of ODI2NDI drivers. You can use this sample file as a skeleton or create your own as described in 3.2.4, "MPTS Response File" on page 58.

Ensure that you have response files for all products you want to install.

12.3.9 Build Client Installation Control Files

A special LCU REXX command file is called from the LCU agent running on the client to control the installation process. How the LCU command files are made and work are explained in detail in 4.4, "LCU Command File" on page 143. There is a sample LCU command file named DEFAULT.CMD provided in the NetWare directory of the sample code CDROM. This sample file is adjusted to install OS/2 V2.11, LAN Adapter and Protocol Support, NetWare requester and additional products. Copy this DEFAULT.CMD to the D:\CIDCLIENT directory and use it as a model. Please take some time to read that section carefully, before editing your own LCU command file(s).

12.4 Preparation of Client Workstations

This section describes the creation of the client boot diskettes that connect a diskette-initiated workstation to the code server. As we are using Novell NetWare as the LAN transport system to establish a connection between client and code server, these diskettes contain images of bootable OS/2 diskettes extended with the necessary NetWare code to connect to a NetWare server. The boot diskettes are prepared by the code server administrator on a workstation running OS/2 V2.x and NetWare Workstation for OS/2 V2.01 and logged in to the NetWare server with supervisory rights.

12.4.1 Running SEDISK

To create the bootable OS/2 diskettes use 15.1.2, "SEDISK" on page 377.

12.4.2 Adding LAN Transport System to Client diskettes

In order to transfer the necessary NetWare code to the LTS diskette, the administrator has to follow the steps described below:

1. Insert the LTS diskette and copy the following files from the D:\CIDIMGNETWARE subdirectory to the diskette:

- LSL.SYS
- TOKEN.SYS
- IPX.SYS
- NWREQ.SYS
- NWIFS.IFS
- NWDAEMON.EXE
- DDAEMON.EXE
- NWREQOS2.MSG
- LOGIN.MSG
- IPXCALLS.DLL

NWLOCALE.DLL
NETSUB.DLL
NWCALLS.DLL
NWNET.DLL
NWCONFIG.DLL
NETAPI.DLL
UNI_437.001
UNI_850.001
UNI_COL.001
UNI_MON.001
437_UNI.001
850_UNI.001

(ROUTE.SYS: see the following note)

The files with the extension **001** belong to the US English version of NetWare. If you are using a different version take the files with the extension of your NLS table (that is UNI_437.049, 437_UNI.049, 850_UNI.049 and UNI_850.049 for the German version for example).

— **Attention ! Space Restriction on Diskette 1** —

As there is not enough space on the 1.44MB formatted diskette 1 for all files that are needed, we did not include the ROUTE.SYS driver. If you need this driver because you have routers in the network between the server and the client, you have to eliminate another file. This could be the HPFS.IFS if you do not have to support HPFS drives. You could also erase the NWREQOS22.MSG though this leads to lots of

SYS0318: File not found

during the boot process. The best solution would be to use 2.88MB formatted diskettes if the client workstations support those. If you are able to track which of the client workstations are Micro Channel machines and which are AT-Bus machines, you can delete the *01.SYS files on the diskettes for Micro Channel machines and the *02.SYS files for AT-Bus machines.

When using **OS/2 Warp V3** it is enough to delete the files

UNPACK.EXE
UNPACK2.EXE

on diskette 1.

2. From the CIDIMGSAMPLENetWare subdirectory the following file has to be copied to the diskette:

STARTRFI.CMD

3. From the CIDIMGSAMPLE subdirectory the following file has to be copied to the diskette:

CRENVVAR.EXE

4. Changes have to be made to the CONFIG.SYS reflecting the NetWare drivers and the paths needed to attach the CID directory structure. The following figure shows how the CONFIG.SYS has to be changed:

```

BUFFERS=32
IOPL=YES
MEMMAN=NOSWAP
PROTSHELL=SYSINST1.EXE
SET OS2_SHELL=CMD.EXE /K A:\STARTRFI.CMD
DISKCACHE=64,LW
PROTECTONLY=YES
LIBPATH=.;\;\OS2\DLL;X:\DLL\OS2V211;X:\IMG\LCU;
IFS=HPFS.IFS /C:64 /AUTOCHECK:C
PAUSEONERROR=NO
CODEPAGE=850
DEVINFO=KBD,US,KEYBOARD.DCP
DEVINFO=SCR,EGA,VTBL850.DCP
DEVICE=\DOS.SYS
DEVICE=\MOUSE.SYS
SET PATH=\;\OS2;.....;X:\IMG\OS2V211\DISK_1;A;\;
SET DPATH=\;\OS2;.....;X:\IMG\LCU;
.
.
.
.

SET SOURCEPATH=X:\
REM --- NetWare Requester Statements BEGIN ---
DEVICE=A:\LSL.SYS
REM --- Network Adapter Card ---
DEVICE=A:\TOKEN.SYS
REM DEVICE=A:\ROUTE.SYS
DEVICE=A:\IPX.SYS
DEVICE=A:\NWREQ.SYS
IFS=A:\NWIFS.IFS
RUN=A:\NWDAEMON.EXE
RUN=A:\DDAEMON.EXE
REM --- NetWare Requester Statements END ---

```

Figure 74. Modified CONFIG.SYS of Second LTS Diskette. For information on ROUTE.SYS see the note on the previous page.

The **STARTRFI.CMD** supplied on the sample code CDROM is invoked by the CONFIG.SYS. It was created to connect the client to the code server. It prompts the user for Login-name and Login-password and keeps this information in a file called ENV_VARS.CMD using the CRENVVAR.EXE. See Appendix F, "Create Environment Variables Program Description" on page 545 for further information on CRENVVAR.EXE. The file ENV_VARS.CMD is then used for the next login after a reboot. Additionally the STARTRFI.CMD

includes mapping statements for the client and it invokes **SRVREXX** located on the code server. **SRVREXX** is a utility loading REXX support from the code server into client workstation memory in order to run REXX LCU command files. Finally, **STARTRFI.CMD** executes the LCU command file for the client with the name <LOGINNAME>.

```

REM This command file is used to connect the Novell NetWare client workstation
REM to the NetWare Server containing the LCU command file, and required
REM CID images.

REM First check to see if the ENV_VARS.CMD file exists, and if not prompt
REM for a NetWare Login-name and Login-password.
IF EXIST ENV_VARS.CMD GOTO SETVARS
@CRENVVAR /P:"Please enter your Login Name" /v:LOGINNAME
          /P:"Please enter your Login Password" /v:LOGINPASSWORD

:SETVARS
@ECHO OFF

REM The ENV_VARS.CMD file will set Login-name and Login-password into the OS\2
REM Environment, so that it can be used in further login's without prompting
REM the user for this information more than once.
CALL ENV_VARS
@ECHO OFF
:START
IF EXIST L:\OS2\LOGIN.EXE GOTO EXIT
GOTO START
:EXIT
L:\OS2\LOGIN %LOGINNAME% %LOGINPASSWORD%

REM Attaching to the CID Code Server paths on the Novell NetWare server

L:\OS2\MAP X:=NETWARE40\SYS:CID
L:\OS2\MAP L:=NETWARE40\SYS:CID\LOG

REM The SRVREXX.EXE run in detached mode will allow the running of REXX programs
DETACH X:\IMG\LCU\SRVREXX.EXE

REM The client LCU command procedure is executed with the Login-name parameters
REM set by the ENV_VARS.CMD file.

X:\CLIENT\%LOGINNAME% %LOGINNAME% L:\%LOGINNAME%.LOG
EXIT

```

Figure 75. **STARTRFI.CMD** File

The **STARTRFI.CMD** has to be changed to reflect your server name and your mappings if they are different from our example.

If the client workstation is started with the now prepared client diskettes, it will connect to the code server and start installations. Please see 4.2.1, "Installation of Novell NetWare Workstation for OS/2 V2.01" on page 128 for a detailed description how the NetWare requester is installed. Check 4.4.7.4, "NetWare LCU Command File" on page 168 for a description of the LCU command file used for NetWare.

12.5 Running the Code Server

Once the Novell NetWare server is running, clients can connect to it and start installations. There is no need to start an additional function on the server for the code server.

In the connection information of the MONITOR function of the NetWare server V.4.0 the actually connected clients and their file access can be monitored.

In order to prevent unauthorized access to files during an install the location of the CID directory structure should be carefully chosen. Especially when granting the permissions to the directory tree, the splitting between the LOG area, where CREATEREADWRITE access is needed and the main CID directory where READFILESCAN access is enough should be followed. If there is a space limitation or other reason that do not allow to have READ and READWRITE access on the same (physical) drive on the server, there is the possibility to leave the recommended directory structure and place the LOG area somewhere else. Remember to change the mappings in the STARTRFI.CMD when doing so.

12.5.1 Customizing Client Diskettes

12.5.1.1 Individual LAN Transport Services Diskettes

The tailoring of individual LTS diskettes means as many LTS diskettes need to be created as there are workstations on the LAN. With a large number of installations and workstations this could prove impractical.

The administrator's work consists of creating the appropriate response files and appropriate LCU command files. The file ENV_VARS.CMD has to be added to every diskette including the LOGINNAME and LOGINPASSWORD, if user prompting is to be suppressed. The following figure gives an example of the file ENV_VARS.CMD.

```
SET LOGINNAME=NWCLIENT
SET LOGINPASSWORD=CID
```

Figure 76. ENV_VARS.CMD File for NetWare

It is also a good idea to write the LOGINNAME on the external diskette label.

12.5.1.2 Common LAN Transport Services Diskettes

The prompted version is much easier as only **one** type of LTS diskette exists which could be copied as many times as needed. By having the user type in the workstation name the individuality would come during access to the LCU command file.

In this case the administrator's work would consist of creating only one type of LTS diskette, and the individual LCU command file and response files.

Note: Of course there is a need for one set of LTS diskettes for each type of LAN adapter used for installations.

Chapter 13. Manual Setup of IBM TCP/IP Version 3.0

This chapter describes the setup of a TCP/IP server to install OS/2 and related products for a remote install using the CID installation method. For the software versions that we used at the time of writing please see Appendix B, "Versions Used in This Book" on page 431.

13.1 Overview

The CID installation method uses the concept of redirected drives to gain access to the images that are accessed on a server and a response file that contains all necessary configuration information as input to the installation. There is little or no user intervention required once the process has started. The disk images and the response files will be read from the server. This requires a special setup of the server:

- Creation of the recommended CID directory structure on the server
- Copying of the diskette images onto the server
- Creation of LCU command and response files for the clients
- Creation of the LAN transport system diskettes

In this chapter we will show how to use a TCP/IP server as a CID code server that runs the LAN CID Utility as a tool for the remote install of CID-enabled products using the Network File System** (NFS**) feature of TCP/IP for OS/2 and redirected drives. Use of this facility allows a workstation to be installed from the following systems that provide NFS server capability:

- OS/2 systems (TCP/IP for OS/2 + NFS feature)
- AIX* systems (IBM RISC System/6000*)
- UNIX** systems which support an NFS server capability

For a detailed description on how the LAN CID Utility works, please refer to 4.4, "LCU Command File" on page 143.

13.2 Scenario

In this chapter we will walk you through the manual setup of a code server using TCP/IP V3.0 to install OS/2 Warp Connect, MPTS, and IBM TCP/IP Version 3.0.

It is assumed that the reader is familiar with the TCP/IP and NFS terms and commands. If this is not the case, it is recommended that a copy of the *IBM Transmission Control Protocol/Internet Protocol Version 2 for OS/2: Installation and Administration*, SC31-6075-06 and *IBM Transmission Control Protocol/Internet Protocol Version 2.0 for OS/2: Network File System Guide*, SC31-7069-01 manuals are available.

It is assumed that the TCP/IP server is installed and running.

Overview of installation steps for TCP/IP:

1. Install the TCP/IP and NFS products on the server.
2. Create a code server directory structure.
3. Load OS/2 CID Utilities.
4. Load the product diskette images using the product dependent procedures.
5. Load EXE and DLL files that are used to support the clients during installation.
6. Install the LAN CID Utility.
7. Build product dependent response files.
8. Build LCU command files.
9. Create client boot diskettes for diskette-initiated workstations.
10. Use boot diskettes to start client and initiate installation.

13.3 Manual Installation

13.3.1 Basic Installation of TCP/IP Server

Please refer to the product manuals for information about installing a TCP/IP server.

In our scenario, TCP/IP is installed on the logical C: drive of an OS/2 Warp Connect system using the defaults supplied. The CID directory structure was created on a separate logical Drive H: The OS/2 installation includes REXX support. The NFS option is installed.

The NFS-kit is not a part of TCP/IP V3.0 So you first have to install OS/2 Warp Connect, TCP/IP V3.0 and then separately the NFS kit. **Note:** If you use the NFS kit from IBM TCP/IP Version 2.0 with IBM TCP/IP Version 3.0 you must apply an APAR PN69745. To request this fix, contact IBM Service at 1-800-237-5511 in the U.S. or your local IBM representative.

An IP address of 9.83.140.198 and hostname of "ITSCTCP" was assigned to the server.

Note

You can use a different hostname and IP address. Please remember to change these values in all procedures and command files if you do so.

Remember to change the commands used here to your actual drive letter.

13.3.2 Creating the CID Directory Structure

The recommended common CID directory structure to be used with LCU is described in Chapter 2, "Recommended CID Directory Structure" on page 39. This common directory structure has been developed for the CID process to ensure compatibility/migration between LAN CID Utility, and IBM NetView Distribution Manager/2 Version 2.1. LCU, by itself, does not require any fixed directory structure. However, we recommend the use of the CID common directory structure to avoid any compatibility problems with follow-on products that will be shipped in the future.

Ensure that the disk you want to use has enough free space to hold the desired product images and has additional space available for response, LCU command and log files.

See Table 10 on page 264.

Use **MKDIR** or **MD** to create the directory structure.

13.3.3 Loading OS/2 CID Utilities to Code Server

Please see Chapter 15, "OS/2 CID Utilities" on page 373 on how to load OS/2 CID Utilities to the code server.

13.3.4 Loading Diskette Images

Please see Chapter 16, "Loading Product Images to Code Server" on page 379 on how to load the product diskette images to the code server.

You will need at least the following diskette images to perform a successful install:

- OS/2 Warp Connect. See 16.1.1, "Loading OS/2 Diskette Images with SEIMAGE" on page 379.
- MPTS. See 16.1.2, "Loading LAN Transport System Diskette Image(s) with LAPSDISK" on page 382. You should use the latest available version of MPTS. In our environment we used the MPTS shipped with IBM Operating System/2 Local Area Network Server V5.0. The LAN Adapter and Protocol Support shipped with MPTS was Version 5.00 (CSD level WR08200). The TCP/IP protocol is included in the LAPS shipped with the MPTS of OS/2 Warp Connect
- TCP/IP V3.0. See 16.1.7, "Loading TCP/IP Diskette Images" on page 392.

Note

It is not possible to create TCP/IP boot disks for the client installation with TCP/IP V3.0. The programs used on the client's boot disk to setup the TCP/IP protocol (arp.exe and ifconfig.exe) try to load the PMWIN.DLL when they load into memory. As there is no presentation manager available when booting from diskettes the program load fails. So we had to use the TCP/IP V2.0 Code to create the boot disks. The files used for the boot disks are on the sample CDROM in the sampleimgtcpclt directory.

The installation of TCP/IP is done in two steps:

After the client workstation is booted from diskettes and OS/2 and MPTS are installed, a subset of the TCP/IP files is transferred to the client, and the CONFIG.SYS file is updated to allow the client to reconnect to the code server. This is necessary because the TCP/IP V3.0 install program needs Presentation Manager to be available. The complete install of TCP/IP V3.0 will follow after the first reboot of the client.

For this initial install of TCP/IP, you need to create a separate subdirectory D:CIDIMGTCPCLT.

Use XCOPY to copy the files needed for the boot disks from the sample code CDROM TCPIPTCPIPCLT directory to the H:CIDIMGTCPCLT directory.

```
XCOPY F:\TCPIP\TCPIPCLT\*.* H:\CID\IMG\TCPCLT /S/E/V
```

Finally, you should copy the contents of the sample code CDROM's TCPIP and UTILITY directory to the server.

```
XCOPY A:\TCPIP\ D:\CID\IMG\SAMPLE\TCPIP /V  
XCOPY A:\UTILITY D:\CID\IMG\SAMPLE /V/S/E
```

13.3.5 Copy REXX to Code Server

GETREXX helps you copy the REXX support necessary for the clients to be able to run their command files. See 17.1.2, "GETREXX" on page 398.

13.3.6 Copy SETBOOT.EXE to Code Server

GETBOOT helps you copy SETBOOT.EXE and XCOPY.EXE, which are necessary for the clients to be able to run their command files. See 17.1.1, "GETBOOT" on page 397.

13.3.7 Code Server Installation

As we use the transport mechanism of Network File System to connect clients to the code server no additional server installation is necessary. Clients, however, have to be defined by the administrator, exports have to be prepared for the CID directory structure and permissions have to be given to the clients:

- Read permissions for the CID directory
- Read and write permissions for the CIDLOG subdirectory

These permissions are defined in the EXPORTS file. An example of an EXPORTS file can be found in the TCPIP subdirectory of the sample code CDROM and is shown in the figure below.

```
H:\cid -ro  
H:\cid\log -rw
```

Figure 77. Example EXPORTS File on NFS TCP/IP Server for CID

Note: We had a problem while installing MPTS on the client and fixed it by removing the -ro from the H:CID export entry. The EXPORTS file has to be placed in the C:MPTNETC directory on the code server or to the directory your environment variable **ETC** refers to.

The HOSTS file reflects the correct hostname and IP address of the NFS server that will be used for code distribution. A sample HOSTS file can be found in the TCPIP subdirectory of the sample code CDROM and is shown in the figure below.

```
9.83.140.198 ITSCTCP
9.83.140.201 ITSCWRKM
```

Figure 78. Example HOSTS File

The HOSTS file has to be placed in the C:MPTNETC on the code server. It is also placed in the H:CIDIMGTCPCLTETC subdirectory and on the LTS diskette which is described later in this chapter. This file must be edited if you are not using the same names as used in this scenario. Please change the file in all occurrences.

13.3.8 Build Response Files

Response files and the utilities to create them are explained in Chapter 3, "Response Files" on page 47.

You need at a minimum proper response files for OS/2 Warp Connect, MPTS and TCP/IP V3.0. Please refer to 3.2.7, "TCP/IP Response File" on page 75 for information on the TCP/IP V3.0 response files. Please remember that there is a sample TCP/IP V3.0 response file named TCPSAM.RSP supplied in the H:CIDIMGSAMPLETCPIP directory. Copy this file to your H:CIDRSPTCPIP30 subdirectory and use it as a skeleton. There is also a sample MPTS response file named MPTSTCP.RSP provided in this directory. This sample MPTS response file reflects the special setup needed in the TCP/IP environment. You can use this sample file as a skeleton or create your own as described in 3.2.4, "MPTS Response File" on page 58.

Ensure that you have response files for all products you want to install.

13.3.9 Build Client Installation Control Files

A special LCU REXX command file is called from the LCU agent running on the client to control the installation process. How the LCU command files are created and work are explained in detail in 4.4, "LCU Command File" on page 143. Please take some time to read the chapter carefully, before editing your own LCU command file(s).

A default LCU command file for TCP/IP is provided in the D:CIDIMGSAMPLETCPIP directory. Copy this CONNECT.CMD to the H:CIDCLIENT directory and use it as a model. This CONNECT.CMD is prepared to install more products than those mentioned in this chapter. If you need information about the install of these additional products, please see the corresponding installation and response file chapters of this book.

13.4 Preparation of Client Workstations

This section describes the creation of the client boot diskettes that connect a diskette-initiated workstation to the code server. As we are using the Network File System as the LAN Transport system to establish a connection between client and code server, these diskettes contain bootable OS/2 diskettes extended with the necessary TCP/IP code to connect to a TCP/IP server. The boot diskettes are prepared by the code server administrator on the code server.

13.4.1 Running SEDISK

To create the bootable OS/2 diskettes see 15.1.2, "SEDISK" on page 377.

13.4.2 Adding LAN Transport System to Client Diskettes

In order to transfer the necessary TCP/IP code to the diskette, the administrator has to follow the steps described below.

Space Restriction in OS/2 Warp V3

If you are installing OS/2 Warp V3 there is not enough space on the LTS diskette. Delete the file **UNPACK*.EXE** on the LTS diskette to have sufficient space. We also deleted the **XDFLOPPY.FLT** to get enough space on the diskette. This file is not needed for the installation itself.

1. Insert the LTS diskette (the second diskette that was created by SEDISK) and copy all files from the H:CIDIMGTCPCLT directory on your server to the diskette:

XCOPY of TCP/IP Files to LTS Diskette

```
XCOPY H:\CID\IMG\TCPCLT A:/S/V
```

Check that the HOSTS file copied to A:ETC contains the correct IP address and hostname.

2. The following file has to be copied to the diskette from the H:CIDIMGSAMPLE subdirectory:

CRENVVAR.EXE

3. The following file has to be copied to the diskette from the H:CIDIMGSAMPLETCPIP subdirectory:

NFSRFI.CMD

4. Add the necessary LAPS files to the client diskettes. Use the code server's C:IBMCOM subdirectory as source for these files, assuming that MPTS is installed on the code server's drive C:.

```
COPY C:\IBMCOM\LANMSGDD.OS2 A:
COPY C:\IBMCOM\LANMSGEX.EXE A:
COPY C:\IBMCOM\PROTMAN.OS2 A:
COPY C:\IBMCOM\PROTOCOL.INI A:
COPY C:\IBMCOM\PROTOCOL\NETBIND.EXE A:
COPY C:\IBMCOM\PRO.MSG A:
COPY C:\IBMCOM\LT2.MSG A:
COPY C:\IBMCOM\LTO.MSG A:
COPY C:\IBMCOM\DLL\LANMSGDL.DLL A:
COPY C:\IBMCOM\MACS\IBMTOK.OS2 A:
```

5. As the server's LAPS configuration is used in the preceding step, changes may be necessary:
 - Edit the PROTOCOL.INI copied to the LTS diskette. If locally administered addresses are used, change the address to avoid conflicts. If there are more protocols defined than TCP/IP, remove those protocol definitions.
 - If the client machine has different network adapters than the server system, you have to replace IBMTOK.OS2 with the correct network adapter driver. You also have to change the PROTOCOL.INI accordingly. (If you do not know how to do this, you can configure LAPS on the server system for the client system as described in E.3.1, "Support for Additional Drivers" on page 534.)
6. Changes have to be made to the CONFIG.SYS reflecting the TCP/IP drivers and the paths needed to attach the CID directory structure.

Remember to change the line **DEVICE=IBMTOK.OS2** if you are using different adapters.

```
buffers=32
iopl=yes
memman=noswap
protshell=sysinst1.exe
set os2_shell=cmd.exe /K a:\NFSRFI.CMD
diskcache=64,LW
protectonly=yes
libpath=.;\;os2\dll;x:\dll\connect;x:\img\lcu;
ifs=hpfs.ifs /c:64
pauseonerror=no
codepage=850
devinfo=kbd,us,keyboard.dcp
devinfo=scr,ega,vtb1850.dcp
device=\dos.sys
set path=\;os2;.;x:\img\connect\disk_1;x:\img\connect\disk_2;x:\img\l
cu;
set dpath=\;os2\os2\system\os2\install;x:\img\lcu;
set keys=on
.
.
.
set sourcepath=x:\
SET ETC=A:\ETC
SET TMP=A:\
DEVICE=A:\LANMSGDD.OS2 /I:A:\
DEVICE=A:\PROTMAN.OS2 /I:A:\
DEVICE=IBMTOK.OS2
DEVICE=INET.SYS
DEVICE=IFNDIS.SYS
CALL=A:\NETBIND.EXE
RUN=A:\LANMSGEX.EXE
RUN=A:\CNTRL.EXE
IFS=A:\NFS200.IFS
```

Figure 79. Modified CONFIG.SYS of Second LTS Diskette

The **NFSRFI.CMD** supplied on the sample code CDROM is invoked by the CONFIG.SYS. It was created to connect the client to the code server. It prompts the user for Login-name and Login-password and keeps this information in a file called ENV_VARS.CMD using the CRENVVAR.EXE. See Appendix F, "Create Environment Variables Program Description" on page 545 for further information on CRENVVAR.EXE. The file ENV_VARS.CMD

is then used for the next login after a reboot. Additionally the NFSRFI.CMD includes mapping statements for the client and it invokes **SRVREXX** located on the code server. SRVREXX is a utility loading REXX support from the code server into client workstation memory in order to run REXX LCU command files. Finally, NFSRFI.CMD executes the LCU command file for the client with the name <HOSTNAME> and IP address <ADDRESS>. The following figure shows the NFSRFI.CMD procedure. If you are using this procedure, remember to change the server name **ITSCTCP** to the name you are actually using. Check the **mount** commands if you placed your CID directory structure to another drive than the H: drive.

```

REM This command file is used to connect the TCP/IP client workstation
REM to the TCP/IP Server containing the LCU command file, and required
REM CID images.

REM First check to see if the ENV_VARS.CMD file exists, and if not prompt
REM for a TCP/IP Host name (your machine id) and your machines address in the
REM network.
IF EXIST ENV_VARS.CMD GOTO SETVARS
@CRENVVAR /P:"Enter your Workstation Name" /v:HOSTNAME /P:"Enter your Login Address" /v:ADDRESS

:SETVARS
@echo off

REM The ENV_VARS.CMD file will set the address and host name into the OS/2
REM Environment, so that it can be used in further login's without prompting
REM the user for this information more than once.
CALL ENV_VARS
@echo off
arp -f
ifconfig lan0 %ADDRESS% mtu 2000
detach nfsctl.exe

REM Attaching to the CID code server paths on the TCP/IP NFS Server
mount -u -g x: ITSCTCP:H:\cid
mount -u -g l: ITSCTCP:H:\cid\log

REM The SRVREXX.EXE run in detached mode will allow the running of REXX programs
DETACH X:\IMG\LCU\SRVREXX.EXE

REM The client LCU command file is executed with the host name and address
REM parameters set by the ENV_VARS.CMD file.
X:\CLIENT\%HOSTNAME% %HOSTNAME% L:\%HOSTNAME%.LOG
EXIT

```

Figure 80. NFSRFI.CMD File

If the client workstation is started with the now prepared client diskettes, it will connect to the code server and start installations.

MTU Size

The MTU size of 2000 was proved as a working level in our scenarios while increased values lead to TRAPs.

13.5 Running the Code Server

Once the TCP/IP server is running, clients can connect to it and start installations. Remember to start the NFS function on the server.

Note: Remember that the NFS function requires the **portmap** function to be loaded.

13.5.1 Customizing Client Diskettes

13.5.1.1 Individual LAN Transport Services Diskettes

The tailoring of individual LTS diskettes means as many LTS diskettes need to be created as there are workstations on the LAN. With a large number of installations and workstations this could prove impractical.

The administrator's work consists of creating the appropriate response files and appropriate LCU command files. The file ENV_VARS.CMD has to be added to every diskette including the HOSTNAME and ADDRESS, if user prompting is to be suppressed. The following figure gives an example of the file ENV_VARS.CMD.

```
SET HOSTNAME=ITSCWRKM
SET ADDRESS=128.0.100.31
```

Figure 81. ENV_VARS.CMD File for TCP/IP

It is also a good idea to write the HOSTNAME on the external diskette label.

13.5.1.2 Common LAN Transport Services Diskettes

The prompted version is much easier as only **one** type of LTS diskette exists which could be copied as many times as needed. By having the user type in the workstation name the individuality would come during access to the LCU command file.

In this case the administrator's work would consist of creating only one type of LTS diskette, and the individual LCU command file and response files.

Note: Of course there is a need for one set of LTS diskettes for each type of LAN adapter used for installations.

We added a DELETE statement in the THINTCP.CMD procedure to delete the ENV_VARS.CMD from the boot diskette, so the LTS diskettes can be used by all clients with the same LAN adapter.

13.6 Additional Installation Procedures and Files

In this section, the additional procedures used for the install of TCP/IP V3.0 are described. All procedures can be found in TCPIP subdirectory of the sample code CDROM.

As mentioned before, we use two steps to install TCP/IP V3.0: After the client workstation is booted from diskettes and OS/2 and LAPS is installed, a subset of the TCP/IP files is transferred to the client, and the CONFIG.SYS is updated to allow the client to reconnect to the code server. This is necessary because the TCP/IP V3.0 install program needs Presentation Manager to be available. The install of this "thin" TCP/IP is done by the procedure **THINTCP.CMD**. When the complete TCP/IP client is installed, this temporary client has to be deleted, because of the different versions. This procedure can be found in the TCPIP directory of the sample code CDROM and is copied during the setup of the server to the H:\CIDIMGSAMPLE\TCPIP directory.

The **THINTCP.CMD** procedure is run from the LCU command file and is used to copy the minimal TCP/IP code to a client workstation during initial install. Additionally this command procedure will set up the required environment on the client workstation to reattach to the code server, after OS/2 has been installed, and a reboot done. The following figure shows the THINTCP.CMD procedure.

```

/* ----- */
/* THINTCP.CMD */
/*
/* REXX program which will perform the following steps -
/*
/* 1. XCOPY the necessary TCPIP files from the server to the client.
/* 2. Update existing lines (PATH,LIBPATH,etc.) in the client CONFIG.SYS as
/*    required.
/* 3. Add new lines to the client CONFIG.SYS (LAN drivers,etc.) as well as
/*    environment variable set statements
/*
/* This program is run from the LCU command file. It assumes the LTS diskette
/* will be left in the drive until the end of the RSPINST (in order to copy
/* ENV_VARS.CMD)
/* ----- */

address cmd          /* Set command processor */
'@echo off'
env='OS2ENVIRONMENT' /* get OS2 environment */
hname=value('hostname',,env) /* get environment variable %hostname% */
addr=value('address',,env) /* get environment variable %address% */
Cltdrv = 'C:' /* default OS/2 drive */
rdir='1>nul 2>nul'

/* You could also set a TCPDrv variable if
/* you wanted TCP/IP on a separate drive */

/* ----- */
/* copy the TCPIP Requester files to the client hard drive
/* ----- */
say 'Copying TCP/IP files to the hard drive'
say 'This may take some time, please be patient!'

'md 'Cltdrv'\tcpip'
'c:\os2\xcopy x:\img\tcpclt\*. * 'Cltdrv'\tcpip /s /e' rdir

/* ----- */
/* Change the PATH, LIBPATH, etc of CONFIG.SYS
/* ----- */
oldcs=Cltdrv||'\config.sys'
newcs=Cltdrv||'\config.new'

do while lines(oldcs) /* do until end of file */
  line=linein(oldcs) /* read a line */
  line=translate(line) /* make it all upper case */

  if substr(line,length(line),1) == ';' /* check for semicolon at lineend */
    then sc = ''
    else sc = ';'

  if pos('LIBPATH',line) <> 0 then do /* if we're on the LIBPATH line */
    line=line||sc||Cltdrv||'\TCP\DLL;X:\DLL\CONNECT;X:\IMG\LCU;',
    ||Cltdrv||'\IBMC\COM\DLL;'
  end
end

```

Figure 82 (Part 1 of 3). THINTCP.CMD Procedure

```

if pos('SET PATH',line) <> 0 then do /* if we're on the PATH line */
    pl='\\tcpip;x:\img\connect\disk_1;x:\img\connect\disk_2;x:\img\lcu;'
    line=line||sc||Cltdrv||pl
end
if pos('SET DPATH',line) <> 0 then do /* if we're on the DPATH line */
    line=line||sc||'x:\img\lcu;'||Cltdrv||'\ibmcom;'
end

if pos('SET HELP',line) <> 0 then do /* if we're on the HELP line */
    line=line||sc||Cltdrv||'\tcpip\help;'
end

call lineout newcs,line /* write line to config.new */
end

/* ----- */
/* Now add the new lines required to config.new */
/* The following line adds the TCP/IP drivers, modify if necessary */
/* ----- */
call lineout newcs,'SET ETC='||Cltdrv||'\tcpip\etc'
call lineout newcs,'SET TMP='||Cltdrv||'\tcpip\etc'
call lineout newcs,'TIMESLICE=100,100'
call lineout newcs,'RUN='||Cltdrv||'\tcpip\cntrl.exe'
call lineout newcs,'IFS='||Cltdrv||'\tcpip\nfs200.ifs'

/* ----- */
/* close the files */
/* ----- */
call stream oldcs,'c','close'
call stream newcs,'c','close'

/* ----- */
/* copy the new config.sys into place */
/* ----- */
'copy 'newcs' 'oldcs
'erase 'newcs

/* ----- */
/* copy the required TCP/IP files to re-connect to the CID Server once */
/* the system has been booted from the hard drive */
/* ----- */
'copy x:\img\sample\tcpip\startup.tcp ' Cltdrv'\startup.cmd'
'copy x:\img\sample\tcpip\nfsrfi.cmd ' Cltdrv'\
'copy x:\img\sample\tcpip\tcpseed.cmd ' Cltdrv'\
'copy a:\crenvvar.exe c:\
'copy a:\env_vars.cmd c:\
/* At last the env_vars.cmd must be deleted from the
boot disk, to make the
disk usable by more than one client. */
'del a:\env_vars.cmd'

/* ----- */
/* Change the PATH, LIBPATH, etc of CONFIG.SYS */
/* ----- */
oldcs=Cltdrv||'\config.sys'
newcs=Cltdrv||'\config.new'

```

Figure 82 (Part 2 of 3). THINTCP.CMD Procedure


```

do while lines(oldcs)                /* do until end of file          */
  line=linein(oldcs)                 /* read a line                  */
  line=translate(line)               /* make it all upper case       */
  if substr(it,length(it),1) == ';'  /* check for semicolon at lineend */
    then sc = ''
    else sc = ';'
  if pos('LIBPATH',line) <> 0 then do /* if we're on the LIBPATH line */
    line=line||sc||CltDrv||'\TCP\IP\DLL;X:\DLL\OS2V211;||CltDrv|&v
bar.'\IBMCOM\DLL;'
  end
  if pos('SET PATH',line) <> 0 then do /* if we're on the PATH line    */
    pl='\TCP\IP\BIN;X:\IMG\OS2V211\DISK_1;X:\IMG\OS2V211\DISK_2;X:\IMG\LCU;'
    line=line||sc||CltDrv||pl
  end
  if pos('SET DPATH',line) <> 0 then do /* if we're on the DPATH line  */
    line=line||sc||'X:\IMG\LCU;||CltDrv||'\IBMCOM;'
  end

  if pos('SET HELP',line) <> 0 then do /* if we're on the HELP line    */
    line=line||sc||CltDrv||'\TCP\IP\HELP;'
  end
  call lineout newcs,line             /* write line to config.new     */
end
/* ----- */
/* Now add the new lines required to config.new */
/* The following line adds the TCP/IP drivers, modify if necessary */
/* ----- */
call lineout newcs,'SET ETC=||CltDrv||'\TCP\IP\ETC'
call lineout newcs,'SET TMP=||CltDrv||'\TCP\IP\ETC'
call lineout newcs,'TIMESLICE=100,100'
call lineout newcs,'RUN=||CltDrv||'\TCP\IP\BIN\CNTRL.EXE'
call lineout newcs,'IFS=||CltDrv||'\TCP\IP\BIN\NFS200.IFS'
/* ----- */
/* close the files */
/* ----- */
call stream oldcs,'c','close'
call stream newcs,'c','close'
/* ----- */
/* copy the new config.sys into place */
/* ----- */
'copy 'newcs' 'oldcs
'erase 'newcs
/* ----- */
/* copy the required TCP/IP files to re-connect to the CID Server once */
/* the system has been booted from the hard drive */
/* ----- */
'copy x:\img\sample\tcpip\startup.tcp ' CltDrv'\startup.cmd'
'copy x:\img\sample\tcpip\nfsrcfi.cmd ' CltDrv'\
'copy x:\img\sample\tcpip\tcpseed.cmd ' CltDrv'\
'copy a:\crenvvar.exe c:\
'copy a:\env_vars.cmd c:\
'del a:\env_vars.cmd

```

Figure 82 (Part 3 of 3). THINTCP.CMD Procedure

The complete install of TCP/IP V3.0 will follow after the first reboot of the client. This is done using the INSTALL command supplied by TCP/IP V3.0. Please refer to 4.1.9, "TCP/IP V3.0" on page 123 for information on the syntax. After the complete install of TCP/IP V3.0 the STARTUP.CMD has to be changed. The call to the TCPSTART.CMD is exchanged with the call to the NFSRFI.CMD which is the procedure that provides the reconnection to the code server. The procedure doing this, **TCPCOPY.CMD**, can also be found on the sample code CDROM. This procedure can also be used to provide the client workstation with specific TCPSTART and SETUP procedures. The TCPSTART.CMD and SETUP.CMD we used can also be found on the sample code CDROM. The following figure shows the TCPCOPY.CMD.

```
/*Copy the STARTUP.CMD and specific TCP/IP start files */
Cltdrv='C:'
'copy x:\img\sample\tcpip\startup.tcp' Cltdrv'\startup.cmd'
'copy x:\img\sample\tcpip\tcpstart.cmd' Cltdrv'\TCPIP'
'copy x:\img\sample\tcpip\setup.cmd' Cltdrv'\TCPIP\BIN'
exit
```

Figure 83. TCPCOPY.CMD Procedure

The CONNECT.CMD command file supplied in the TCPIP subdirectory of the sample code CDROM includes the product definition for THINTCP, TCPINST and TCPCOPY is prepared to install them.

A procedure to clean up the CONFIG.SYS and STARTUP.CMD after the last install of an LCU command file is executed is also supplied on the sample code CDROM. It is named **TCDELETE.CMD**. Additionally, it will save the necessary files to reconnect to the code server. The following figure shows it.

```

/* ----- */
/* TCDELETE.CMD */
/* ----- */
/* This file will delete the files used to install the TCP/IP Requester */
/* ----- */
address cmd
'@echo off'
Cltdrv='C:' /* Default OS/2 Drive */

'md c:\tcpseed' /* Create the TCP/IP seed directory */

/* ----- */
/* Remove "Attach to CID server", "Startup TCP/IP" command files from the */
/* root directory and copy all code to be reused into the TCPSEED subdir. */
/* ----- */
'copy c:\nfsrfi.cmd c:\tcpseed'
'copy c:\startup.cmd c:\tcpseed\startup.tcp'
'copy x:\img\sample\crenvvar.exe c:\tcpseed'
'del c:\env_vars.cmd'
'del c:\nfsrfi.cmd'
/* ----- */
/* Delete all the CAS statements from the config.sys file */
/* ----- */
do while lines(Cltdrv||'\config.sys') /* do until end of file */
  it = linein(Cltdrv||'\config.sys') /* read first line */
  it = translate(it) /* make everything UPPERCASE */

  if pos('SET CAS',it) <> 0 then do /* read config.sys file and look */
    it='' /* for lines with CAS */
    end /* mark all lines with CAS to blank */

/* ----- */
/* remove the blank lines from the config.sys file */
/* ----- */
if it <> '' then do
  call lineout Cltdrv||'\config.new', it /* Write line to config.new */
end
end
/* ----- */
/* close the files */
/* ----- */
call stream Cltdrv||'\config.new', 'c', 'close'
call stream Cltdrv||'\config.sys', 'c', 'close'
'copy' Cltdrv||'\config.new' Cltdrv||'\config.sys'
'del 'Cltdrv||'\config.new'

```

Figure 84 (Part 1 of 2). TCDELETE.CMD Procedure

```

/* ----- */
/* remove call to temp. TCP/IP startup file from the startup.cmd file */
/* ----- */
do while lines(Cltdrv||'\startup.cmd') /* Do until end of file */
  it = linein(Cltdrv||'\startup.cmd') /* Read first line */
  it = translate(it) /* Make everything UPPERCASE */

  if pos('CALL NFSRFI',it) <> 0 then do
    it='CALL TCPSTART.CMD'
  end

  if it <> '' then do
    call lineout Cltdrv||'\startup.new', it /* Write line to config.new */
  end
end
/* ----- */
/* close the files */
/* ----- */
call stream Cltdrv||'\startup.new','c','close'
call stream Cltdrv||'\startup.cmd','c','close'
'copy' Cltdrv||'\startup.new' Cltdrv||'\startup.cmd'
'del' Cltdrv||'\startup.new'

/* ----- */
/* Save the config.sys file to be used for re-connection to CID server with */
/* TCP/IP connections */
/* ----- */
'copy c:\config.sys c:\tcpseed\config.tcp'

```

Figure 84 (Part 2 of 2). TCDELETE.CMD Procedure

The **TCPSEED.CMD** procedure is used to re-attach the client workstation to the code server using the procedures and files found in the TCPSEED subdirectory.

```

/* ----- */
/* TCPSEED.CMD */
/* ----- */
/* Create attach to TCP/IP Server for SEMAINT */
/* ----- */
'copy c:\config.sys c:\tcpseed\config.os2' /* Change files around */
'copy c:\startup.cmd c:\tcpseed\startup.os2'
'copy c:\autoexec.bat c:\tcpseed\autoexec.os2'
'copy c:\tcpseed\config.tcp c:\config.sys'
'copy c:\tcpseed\nfsrfi.cmd c:\nfsrfi.cmd'
'copy c:\tcpseed\startup.tcp c:\startup.cmd'
'copy c:\tcpseed\crenvvar.exe c:\'
'c:\os2\setboot /ibd:c'

```

Figure 85. TCPSEED.CMD Procedure

The **STARTUP.TCP** file contains the call to the **NFSRFI.CMD** file.

The **TCPREP.CMD** procedure will copy the files required to reattach the TCP/IP client to the code server after SEMAINT has completed.

```

/* ----- */
/* TCPREP.CMD */
/*
/* This REXX procedure which will change the CONFIG.SYS that was created by
/* SEMAINT in order to connect back to the code server after booting from
/* the C:\SERVICE subdirectory.
/*
/* This procedure is invoked from the LCU command file.
/* ----- */

address cmd /* Set command processor */
'@echo off'
env='OS2ENVIRONMENT' /* get OS2 environment */
hname=value('hostname',,env) /* get environment variable %hostname% */
addr=value('address',,env) /* get environment variable %address% */
CltDrv = 'C:' /* default OS/2 drive */
rdir='1>nul 2>nul'

/* You could also set a TCPDrv variable if
/* you wanted TCP/IP on a separate drive */

/* ----- */
/* Change the PATH, LIBPATH, etc of CONFIG.SYS
/* ----- */
oldcs=CltDrv||'\config.sys'
newcs=CltDrv||'\config.new'

do while lines(oldcs) /* do until end of file */
  line=linein(oldcs) /* read a line */
  line=translate(line) /* make it all upper case */

  if substr(line,length(line),1) == ';' /* check for semicolon at lineend */
    then sc = ''
    else sc = ';'

  if pos('SET OS2_SHELL',line) <> 0 then do /* If SET OS2_Shell is in line */
    line=line '/' K C:\NFSRFI.CMD' /* add call for nfsrfi.cmd */
  end

  if pos('LIBPATH',line) <> 0 then do /* if we're on the LIBPATH line */
    line=line||sc||CltDrv||'\TCP\DLL;X:\DLL\OS2V211;'||CltDrv||&v
    bar.'\IBMCOM\DLL;'
  end

  if pos('SET PATH',line) <> 0 then do /* if we're on the PATH line */
    p1='TCP\BIN;X:\IMG\OS2V211\DISK_1;X:\IMG\OS2V211\DISK_2;X:\IMG\LCU'
    line=line||sc||CltDrv||p1
  end

  if pos('SET DPATH',line) <> 0 then do /* if we're on the DPATH line */
    line=line||sc||X:\IMG\LCU;'||CltDrv||'\IBMCOM;'
  end
end

```

Figure 86 (Part 1 of 2). TCPREP.CMD Procedure

```

if pos('SET HELP',line) <> 0 then do /* if we're on the HELP line */
    line=line||sc||CltDrv||'\TCP\HELP;'
end

call lineout newcs,line /* write line to config.new */
end

/* ----- */
/* The following line adds the TCP/IP drivers, modify if necessary */
/* Add also the necessary LAPS statements */
/* ----- */
call lineout newcs,'SET SOURCEPATH=X:'
call LINEOUT newcs,'REM --- TCP/IP Requester Statements BEGIN ---'
call LINEOUT newcs,'DEVICE=||CltDrv||'\IBMCOM\LANMSGDD.OS2 /I:||CltDrv||'\IBMCOM'
call LINEOUT newcs,'RUN=||CltDrv||'\IBMCOM\LANMSGEX.EXE'
call LINEOUT newcs,'REM --- Network Adapter Card ---'
call LINEOUT newcs,'DEVICE=||CltDrv||'\IBMCOM\PROTMAN.OS2 /I:||CltDrv||'\&bsl. IBMCOM'
call LINEOUT newcs,'DEVICE=||CltDrv||'\IBMCOM\MACS\IBMTOK.OS2'
call LINEOUT newcs,'DEVICE=||CltDrv||'\IBMCOM\PROTOCOL\INET.SYS'
call LINEOUT newcs,'DEVICE=||CltDrv||'\IBMCOM\PROTOCOL\IFNDIS.SYS'
call LINEOUT newcs,'RUN=||CltDrv||'\IBMCOM\PROTOCOL\NETBIND.EXE'
call lineout newcs,'SET ETC=||CltDrv||'\TCP\ETC'
call lineout newcs,'SET TMP=||CltDrv||'\TCP\ETC'
call lineout newcs,'TIMESLICE=100,100'
call lineout newcs,'RUN=||CltDrv||'\TCP\BIN\CNTRL.EXE'
call lineout newcs,'IFS=||CltDrv||'\TCP\BIN\NFS200.IFS'

/* ----- */
/* close the files */
/* ----- */
call stream oldcs,'c','close'
call stream newcs,'c','close'

/* ----- */
/* copy the new config.sys into place */
/* ----- */
'copy 'newcs' 'oldcs
'erase 'newcs

```

Figure 86 (Part 2 of 2). TCPREP.CMD Procedure

Chapter 14. Manual Setup of NetView Distribution Manager/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 though you need NetView DM/2 V2.1 if the server is running OS/2 Warp V3.

The enhancements of Version 2.1 are mainly in the new function remote administrator. The tasks for preparing a server workstation to initiate and control installations remained unchanged, though you might receive slightly different messages. On diskette #25 of NetView DM/2 V2.1 there are sample change file profiles, response files and procedures. You might want to check on those samples before creating your own files.

Be sure that you have the latest version, CSD and FixPaks available. Make sure that you applied the latest available CSD/FixPak for the NetView DM/2 images as described in 14.4.3, "Loading Diskette Images" on page 352. Be aware that there are fixes to be applied on top of the FixPak XREFP01 (This FixPak changes the syslevel of NetView DM/2 V2.1 to XR00003). These fixes are PTR0496F, PTR0518F and II08983. If you are in an host connected environment (including NetView DM/MVS) the fix PTR0518F is recommended. If you are using DB2/2 V2.11 you need PTR0496F.

14.1 NetView DM/2 Overview

This section does not explain the architecture, functions or commands of NetView DM/2. It is not the aim of this section to replace the manuals and the practical experience with the product NetView DM/2. Therefore, it is recommended that the administrator has the NetView DM/2 manuals available (and, of course makes use of them!).

Before outlining the process in sequence, it might first be beneficial to review the roles of the stations involved in using NetView DM/2 in a stand-alone LAN environment. Basically these stations can be grouped into two areas:

- NetView DM/2 Change Control Server

This workstation holds the product images and response files. It also processes the change files which 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 also set up as a CC server, prepares all images, response files and change files for the CC server. All code is then sent to the actually executing CC server, using an APPC session.

In a stand-alone LAN environment, NetView DM/MVS is not involved. The preparation site workstation is usually the same system as the NetView DM/2 CC server.

- NetView DM/2 Change Control 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.

The following figure shows the stand-alone LAN environment.

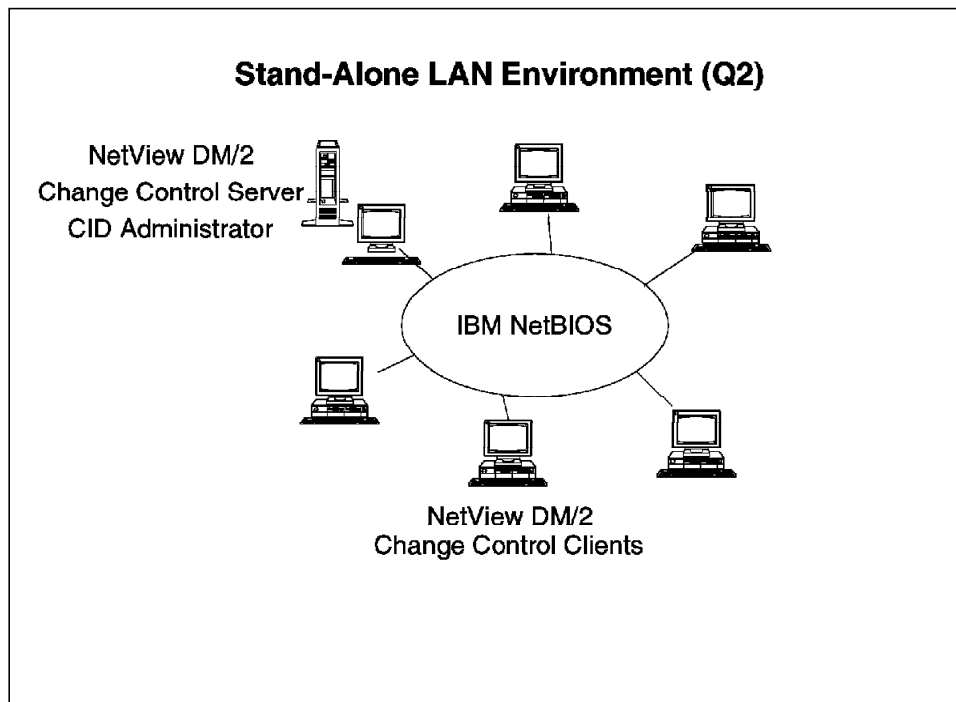


Figure 87. Stand-Alone LAN Environment

14.2 Scenario

In this scenario, a client workstation is installed with OS/2 Warp Connect, MPTS and NetView DM/2 V2.1 client.

The methodology used in this document assumes that the NetView DM/2 client code is NOT installed on the client workstation initially. The client workstation is booted from two NetView DM/2 boot diskettes to establish connectivity with the NetView DM/2 CC server. The server can then issue the command to install OS/2 Warp Connect, MPTS, and NetView DM/2 extended client code using response files and product images stored on the CC server. This is referred to as a *lightly attended* install since user intervention is required to insert and remove the boot diskettes at the client workstation. This methodology is applicable for installing OS/2 on a pristine system or migrating OS/2 if the CC client has not been installed. The same change files can be used to implement both scenarios but with different response files.

After the OS/2 Warp Connect, MPTS and NetView DM/2 installs are completed, the client workstation will reboot. Since the NetView DM/2 client feature is now installed on the client workstation, a session with the NetView DM/2 CC server is automatically initiated by STARTUP.CMD. This enables the CC server to perform *unattended* installs such as Service Pak, migration to later software versions and installing other OS/2 CID enabled applications on the client.

14.3 Overview of Installation Steps for NetView Distribution Manager/2 CC server

1. Install OS/2, MPTS and DB2/2 on the code server.
2. Create CID directory structure and load the product diskette images.
3. Install NVDM/2 Extended Base Package on the server.
4. Prepare the response files.
5. Prepare the change file profiles for the change files that will install the products on the CC clients.
6. Build the change files from the change file profiles.
7. Create the two boot diskettes for diskette-initiated workstations.
8. Define the CC clients.
9. Start the code server.

10. Start the client with the boot diskettes to connect to the server.
11. Install products on the client using the change files.

14.4 Manual Installation

14.4.1 Preparation of Basic Code on the Code Server

The following base software should have been previously installed on the server:

- OS/2 Warp Connect
- MPTS LAN Adapter and Protocol Support
Make sure to use the latest version.
- DB2/2 V2.11
- Communications Manager/2 for APPC connection, if required

For the versions we used, please refer to Appendix B, “Versions Used in This Book” on page 431.

14.4.2 Creating the CID Directory Structure

The common CID directory structure is used to create separate IMG, CSD, RSP and LOG directories for storing the product images, the Service Pak images, product response files and installation logs. Individual product subdirectories are created under each of these directories.

For NetView DM/2, the product images, response files, CSDs and log files are stored in either the SharedDirA (SHAREA) or SharedDirB (SHAREB) directories on the CC server. You can also use CID as directory name instead of SHAREA and SHAREB, as long as the parameters in the IBMNVDM2.INI file are defined correctly. Please see Chapter 2, “Recommended CID Directory Structure” on page 39 for a detailed description and for detailed recommendations when using NetView DM/2.

14.4.3 Loading Diskette Images

Please see Chapter 16, “Loading Product Images to Code Server” on page 379 for how to load the product diskette images to the NetView DM/2 code server.

For this scenario, the following are used on the code server:

- OS/2 Warp Connect. See 16.1.1, “Loading OS/2 Diskette Images with SEIMAGE” on page 379.

- LAN Adapter and Protocol Support. See 16.1.2, “Loading LAN Transport System Diskette Image(s) with LAPSDISK” on page 382.
- NetView DM/2. See 16.1.6, “Loading NetView DM/2 Diskette Images” on page 391.

Additionally, you should copy the contents of the sample code CDROM. On the sample code CDROM, we provide sample profiles that can be used as models.

COPY E:\NVDM2*.pro D:\PROFILES

There are also sample response files for NetView DM/2 provided in the NVDM2 directory of the sample code CDROM. Copy those in the RSPNVDM21 directory.

14.4.4 Loading OS/2 CID Utilities to Code Server

Refer to Chapter 15, “OS/2 CID Utilities” on page 373 for how to unpack and load the OS/2 CID Utilities into the directory structure used for the NetView DM/2 scenarios in this book.

Please remember that the directory where you place the OS/2 utilities can vary. We are using the EXE subdirectory, but many other publications use the IMGCONNECT directory. This affects the creation of the change files where the correct path to the utilities has to be stated.

14.4.5 Code Server Installation

14.4.5.1 Installation of NetView DM/2 on the CC Server

The following section describes the procedure for installing the NetView DM/2 V2.1 extended base package interactively, using the product images, placed in D:SHAREAIMGNVDM21.

Note: Replace D: with whatever drive letter you are using.

1. It is necessary to logon and start database manager as the NetView DM/2 database will be built during the install.

Issue the following commands:

**logon /L userid /p:password
STARTDBM**

2. Backup the essential files.

This step is optional but note that the NetView DM/2 installation program modifies CONFIG.SYS and STARTUP.CMD.

Issue the following commands:

```
C:
CD \
COPY CONFIG.SYS CONFIG.SAV
COPY STARTUP.CMD STARTUP.SAV
```

Note: NetView DM/2 automatically updates the STARTUP.CMD file with the **LOGON** command and the default /L userid /p:password, upon installation. You may want to **REMove** this line if you have a particular logon procedure or wish to log on to the database manually. Even if you are using a DB2/2 version other than the English versions, this will be added.

3. Install NetView DM/2 from the images on the D: drive.

The NetView DM/2 diskette images should have been copied to the D:SHAREAIMGNVDM21 subdirectory in 14.4.3, "Loading Diskette Images" on page 352. Enter the following commands:

```
D:
CD D:\SHAREA\IMG\NVDM21
NVDMPMS
```

4. At the NetView DM/2 base and server features installation screen, press **Enter** to continue; click on **OK**. On the Installation Initialization screen click on **CONTINUE**.
5. Define the installation parameters; these will be stored in the **IBMNVD2.INI** file in your target directory.

At the NetView DM/2 Base and Feature Installation Screen the parameters are:

```
Target Directory - D:\IBMNVD2
Installation Type - Full Installation
Configuration Option - Boot Drive = C:
Connection Type - Standalone
Install Option - CDM only
```

- Select **Configure** to define configuration parameters:

```
Run Time Logging Options - NetView DM/2 Facilities
ServerName - <server name>
MaxClient - specify your own or use 10
Agent Timeout - -1
Adapter Number - the adapter you configured for NetBios
```

- Select **Shared Dirs** to change the share directory path name to the following:

Error Log File - D:\IBMNVDM2\ERROR.DAT
Message Log File - D:\IBMNVDM2\MESSAGE.DAT
Shared Dir A: - D:\SHAREA
Shared Dir B: - D:\SHAREB
File Service Dir - D:\IBMNVDM2\FSDATA

- Select **OK** to return to configure menu.
 - Select **OK** to return to main menu.
 - Select **Install** to install.
 - NetView DM/2 install program progression bar will be displayed.
 - "Installing from hard disk" message will appear.
 - When the installation is completed, select **Continue** to end the install program.
6. Modify the IBMNVDM2.INI file to give the SharedDirA directory read access only:

EPM D:\IBMNVDM2\IBMNVDM2.INI

After the editor comes up, change the SharedDirA statement to the following:

SharedDirA=D:\Sharea,R

Press **< F 4 >** to save and exit.

7. Check the NetBIOS resources of the server with 14.10.2, "NetBIOS Considerations" on page 371.
8. Check if the default LOGON that was entered by NetView DM/2 to the STARTUP.CMD fits your setup. Reboot the workstation.
9. When the system is up again, check the status of the NetView DM/2 components; enter:

CDM STATUS

All components should be active. See 14.8.1, "NetView DM/2 Command Interface" on page 367 for a detailed description of the line commands and their results.

14.4.6 Build Response Files

CID enabled OS/2 applications are installed using a redirected drive and a response file. The response file is a flat ASCII file containing configuration/installation parameters ordinarily entered at screen prompts; these parameters are defined in the response file through keywords. Individual products have defined their own values. Most products come with sample response files on their diskettes. These samples can be used to create your own customized response file.

See Chapter 3, "Response Files" on page 47 for more information on response files.

Remember to create response files for every product you want to install.

14.4.7 Build Change Files

Change files tell NetView DM/2 what action is to take place on the target workstations. They reflect the objects to be sent and the installation procedures to be used. NetView DM/2 uses these change files that are built for the distribution of software and/or files.

See 4.6, "NetView DM/2 Change Control Files" on page 171 for how to create the change file profiles that build these client change files.

Remember to create change files for every product you want to install.

14.5 Preparation of Client Workstations

14.5.1 Creation of Boot Diskettes

Connectivity between the CC server and CC client is established through two boot diskettes which temporarily load a minimal NetView DM/2 CC client system on the workstation.

The same boot diskettes can be used on a pristine machine or on an existing OS/2 workstation without the NetView DM/2 CC client installed. Once the connection with the CC server is established, the CC client can accept commands from the CC server to install various software.

The following utilities are used in creating the boot diskettes:

- SEDISK - Creates minimal OS/2 bootable diskettes.
- THINLAPS - Adds minimal LAN transport system to the boot diskettes.

- NVDMBDSK - Adds NetView DM/2 CC client code to the boot diskettes. This is a PM program. With CSD level XR20466 (leading to syslevel XR00002), there is also a command line utility called NVDMRDSK available. See the README file of the CSD for information on the syntax.

1. Create two minimal OS/2 system boot diskettes.

SEDISK was copied to the CC server in step 14.4.4, "Loading OS/2 CID Utilities to Code Server" on page 353.

Issue the following commands:

D:\SHAREA\EXE\CONNECT\SEDISK /S:D:\SHAREA\IMG\CONNECT /T:A:

Insert a formatted diskette into A: drive and press Enter when prompted.

At completion of the first diskette, remove it and label it "**NetView DM/2 boot diskette #0**"

Insert a new formatted diskette (second diskette) into the A drive when prompted and press Enter. When the SEDISK is completed, a message will be displayed.

2. Install LAN transport system to the boot diskette.

Leave the second bootable diskette in the A: drive and issue the following command:

D:\SHAREA\IMG\MPTS\THINLAPS D:\SHAREA\IMG\MPTS A: IBMTOK.NIF

This adds LAN Adapter and Protocol Support to the second of the two boot diskettes. The THINLAPS program provides the adapter driver for the token-ring adapter, and the NetBIOS protocol drivers. NetBIOS is required by the NetView DM/2 CC client code to establish a session with the CC server via LAN. For more information about THINLAPS, please refer to 4.1.2.3, "THINLAPS" on page 92.

When THINLAPS is completed, you will receive a "**THINLAPS completed successfully**" message.

3. NVDMBDSK installs a minimal NetView DM/2 CC client on the bootable diskette. It is located on an installed NetView DM/2 CC server system in the **IBMNVDM2BIN** directory.

To create the minimal CC client leave the second bootable diskette in drive A: and enter:

D:\IBMNVDM2\BIN\NVDMBDSK

The NetView DM/2 **Boot Diskette Update** window appears showing the following parameters:

Target Environment	Operating System of the client that is to be installed.
Target Drive	Drive where the diskette is inserted.
Server Name	Name of the CC server to which the pristine workstation will be connected. By entering a "?" you will be prompted to type the name of the server during the boot sequence.
Client Name	CC client name of the pristine workstation. By entering a "?" you will be prompted to type the name during the boot sequence.
Attach Timeout	This value specifies how long the CDM agent on the pristine workstation will wait after trying to attach to the CC server. <ul style="list-style-type: none"> • -1 disables the timeout.
Receive Timeout	This value specifies how long the CDM agent on the diskette-initiated workstation will wait for a reply when accessing the CC server shared disk area. If this timeout is exceeded, the CDM agent assumes CC server is inactive and shuts itself down. <ul style="list-style-type: none"> • -1 disables the timeout.
Request Timeout	This value specifies how long the CDM agent on the pristine workstation will wait for a request before terminating. <ul style="list-style-type: none"> • -1 disables the timeout. • 0 terminates immediately if no further requests are pending.
Adapter Number	Adapter number to be used when connecting to the LAN.

Enter the following values:

```

Target Environment = OS/2
Target Drive = A:
Server Name = ?
Client Name = ?
Attach Timeout = -1
Receive Timeout = -1
Request Timeout = -1
Adapter Number = 0

```


Leaving the server and client name as "?" causes NetView DM/2 to prompt the user for the server and client names when the boot diskettes are used. This allows the same set of boot diskettes to be used for multiple workstations. However, the user must know the correct names to enter because the CC client name must be defined to the CC server. If an unknown CC client name is entered, connection to the CC server will be rejected.

Click on **OK** and observe the message:

"The diskette has been updated correctly!".

Select **EXIT** to exit.

Select **YES** to confirm exit.

4. Remove the diskette from the A drive and label it **"NVDM/2 boot diskette #1"**. You now have the two NVDM/2 bootable diskettes.

At the completion of SEDISK, THINLAPS and NVDMBDSK, the following CONFIG.SYS is on NetView DM/2 boot diskette #1:

```

***NVDM/2 additions are highlighted***

buffers=32
iop1=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;
ifs=hpfs.ifs /c:64
.
.
set path=\;\os2;\os2\system;\os2\install;\IBMNVDM2\BIN;Z:\BIN;
set dpath=\;\os2;\os2\system;\os2\install;\IBMNVDM2;Z:\;
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 88. CONFIG.SYS on "NVDM/2 DISK #1" after NVDMBDSK

In addition, **disk #1** was also updated with the directory structure:

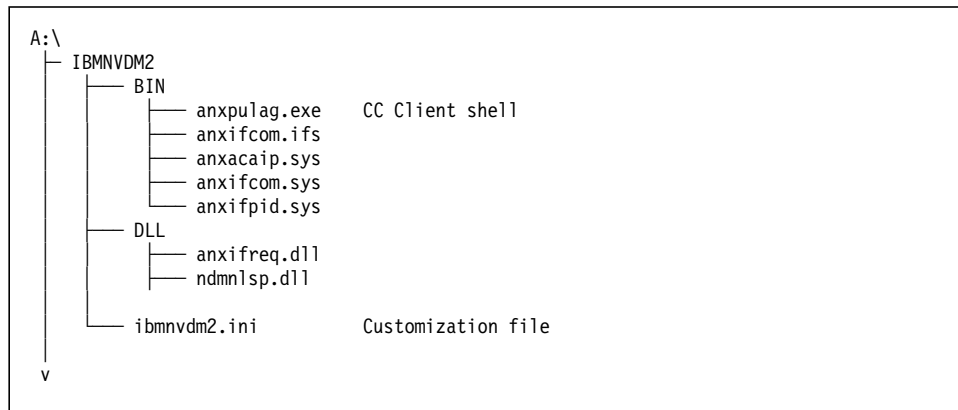


Figure 89. NetView DM/2 Client Directory Structure on NetView DM/2 Boot Disk #1

14.5.2 Connectivity between CC Server and CC Client

14.5.2.1 Define CC Clients and Connect to CC Server

This chapter describes the procedure to establish connectivity between the CC server and the CC client.

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 procedure for creating the NetView DM/2 CC client boot diskettes was described in the previous section of this chapter, 14.5.1, “Creation of Boot Diskettes” on page 356.

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 client is not started, it will have an **inactive** status. Notice that the CC server's name is also 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 is inactive, enter:

CDM START

All NetView DM/2 components on the CC server should be started automatically by STARTUP.CMD. 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.

At the CC client workstation:

3. Start NetView DM/2.

If NetView DM/2 client code is NOT installed:

- Insert NetView DM/2 boot diskette #0 into the A: drive and reboot the workstation.
- When prompted, insert NetView DM/2 boot diskette #1:.
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 "ANX1317W CDM Agent Starting. You can remove boot diskette and leave the workstation unattended" message.
- Remove the boot diskette.
- When the client is attached to the server, you will receive the "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.CMD. If the CDM agent is not active, then issue:

CDM START

At the CC Server:

4. Verify that the CC client is attached to the CC server.

Issue the command:

CDM LIST_WS

If client workstation is booted from diskettes, status should be RUNNING. If NetView DM/2 is already installed on client, then it will have a status of ACTIVE.

14.6 CC Client Operational States

To display the status of the CC clients, you may either issue the CDM LIST_WS command or use the NetView DM/2 dialog to display the CC domain window. There are three operational states for a CC client: inactive, active, running.

<i>Table 13 (Page 1 of 2). CC Client State</i>			
CC client state (as listed by LIST_WS command)	Dialog appearance (in CC Domain)	Description	Maximum clients in this state
Inactive	Dashed Outline	CC client has not been started (i.e., no CDM START has been issued).	1000
Active	Solid Outline	CC client has been started (i.e., a CDM START has been issued).	100

Table 13 (Page 2 of 2). CC Client State			
Running	Solid Outline, Blue Filled	CC Client is processing a Change Management Request (i.e., a CDM INSTALL or ACTIVATE or REMOVE or INITIATE).	10

Note: Refer to the *IBM NetView Distribution Manager/2 Version 2.1 User's Guide*, SH19-5048-02 and the README file on diskette #9 of the NetView DM/2 package for further information. The README.TXT file contains updates to the NetView DM/2 manuals of December 1993. Please note that the limit of 1,000 clients defined per node is the architectural limit of the product. The actual limit for acceptable performance will depend on many factors (such as the size/speed/memory configuration of the system, what other activities are taking place, how many installations are taking place at a time, etc.). IBM does not warrant or imply that any actual production NetView DM/2 server can or will support 1,000 defined clients with acceptable performance or operation.

Important

If a client workstation is booted from the NetView DM/2 boot diskettes, its operating state will be RUNNING regardless of whether a change management request is being processed or not.

14.7 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. For a detailed description on how to create the change files, please see 4.6.3, "Create Change Files to Install CID-Enabled Products" on page 176. You will need change files for OS/2 Warp Connect, MPTS and NetView DM/2 client feature. The following steps show you how to install these change files.

1. Verify that the CC client is attached to the CC server.
This can be done via command line 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:

IBM.OS2.300.CONNECT.INST.REF.1
IBM.MPTS.500.INST.REF.1
IBM.NDM.210.CLIENT.INST.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 all the selected objects listed.

4. Define the installation order.

Click on **Order** to display the Install Order window.

Select **IBM.OS2V30.CONNECT.INST.REF.2.11**.

Select **Up** to move it up to the first position.

Move other objects to arrange in this order:

IBM.OS2.300.CONNECT.INST.REF.1
IBM.MPTS.500.INST.REF.1
IBM.NDM.210.CLIENT.INST.REF.1

Select **OK** to return to the Install window.

5. Define installation options.

Select **Options** from the Install window. The Options screen will be displayed.

IMPORTANT

Select Install as a coreq group.

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. Then the system is rebooted.

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.

6. Select the target workstation for the change files.

Select **client's name** from the workstation list to install the objects on client workstation.

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.

7. 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.

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 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.

2. To install the change files:

```
CDM INSTALL IBM.OS2.300.CONNECT.INST.REF.1
+IBM.MPTS.500.INST.REF.1
+IBM.NDM.210.CLIENT.INST.REF.1 /WS:<client name>
```

Note: Although this command is listed in three lines, it 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 IBMNVDM2 directory or the NVDM/2 manuals for further information on the line commands.

14.8 Operating the Code Server

14.8.1 NetView DM/2 Command Interface

Several commands are available to control NetView DM/2. They can be entered at an OS/2 command prompt or through the panels of the dialog interface. All commands are also available in the dialog interface through 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*, SH19-5048-02 and in the CDM.INF that can be accessed using the VIEW command of OS/2 and is located in the IBMNVDM2 directory.

Some of the global commands which start and stop NetView DM/2 or its subcomponents:

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, as 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.

14.9 Problem Determination

NetView DM/2 has different ways to indicate error conditions:

- Pop-up windows
- Messages and codes
- Error log
- Logging files
- Traces

Note: Refer to the *IBM NetView Distribution Manager/2 Version 2.1 Messages and Error Recovery Guide*, SH19-6924-05 for information on error messages and recovery information.

Also see *Automated Installation for CID Enabled Products Using NetView DM/2 V2.0 and NetView DM R4*, GG24-3782, Chapter "NetView DM/2 V2.0 Problem Determination".

A very helpful way to manage problems in the NetView DM/2 environment is to send a command line to a client. Using this method makes it possible to look at the client's local drives and to invoke OS/2 commands. Even the program invocation of CID enabled products can be tested in this way. If the program is an own written REXX script and the script has typical REXX errors

this is one of the most powerful ways to test the scripts in the environment where they are to be used.

To make a command line available at the CC client you first have to create a changefile profile for the **CMD.EXE**.

```
TargetDir=C:\
Section Catalog
Begin
    ObjectType = SOFTWARE
    GlobalName = ITSO.CMD.REF.1
    Description = Commandline
End

Section Install
Begin
    Program = SA:\EXE\OS2V300\CMD.EXE
    Parms = /K
End
```

Figure 90. Changefile profile for command line

Create a changefile using the **cdm build** command as described above. Once you have created the changefile you can install this command line to any OS/2 client connected to the server. The command prompt is not directly visible when it is installed. You have to switch to the **CDM Agent** to access it. Logical drives connected to the following the NetView DM/2 server areas available:

- SHAREB assigned as W:
- SHAREA assigned as X:
- FSDATA assigned as Y:
- Workstation directory on the CC server ibmnvdm2cmreq<Workstation Name> assigned as Z:

The drive letters mentioned above are the default drives but you can configure them in the CC Server configuration.

Note

If you enter *EXIT* in the command line the command line will be terminated. On the CC server you get the following message:

ANX0061: (E) The INSTALL request for an object with Global Name '****' on workstation '****' was unsuccessful or disabled by the local user.

14.10 Customizing the Code Server

This section will provide an overview of some of the steps/considerations required for:

- User profile management (for use with Database Manager)
- NetBIOS support

14.10.1 User Profile Management (UPM) Considerations

User profile management provides security in OS/2 by defining user IDs with various levels of authority. These levels are:

- Administrator
- Local administrator
- User

Database Manager also defines levels of authority. The relevant levels for NetView DM/2 are:

- *SYSADM* - Highest level of authority. Allows you to create or erase databases, grant access to databases and change database configuration files.
- *DBADM* - Allows you to access and control an existing database.

If you have Local Administrator or Administrator status in User Profile Management then you are automatically given *SYSADM* authority for Database Manager databases.

When using NetView DM/2, you need to be logged on as a:

- *SYSADM* to install NetView DM/2
- *USER* to start NetView DM/2
- *DBADM* to maintain existing databases
- *SYSADM* to create or erase databases

14.10.2 NetBIOS Considerations

NetBIOS is the protocol used to support sessions between a CC server and CC clients. NetView DM/2 V2.1 at syslevel XR00003 and higher uses the following NetBIOS resources (in addition to those required by other applications you have installed):

	CC Server	CC Client
Sessions	2 + MaxClients	2
Commands (NCBs)	29 + MaxRequests	14
Names	7	6
(GDT) Selectors	System Total of 30	Default
Datagram packets	15	Default
NetBiosTimeout	2000	Default
NetBiosRetries	15	Default
Note: The last two values should be used when connecting more than 20 clients to one CC server.		

These resources are defined in PROTOCOL.INI. If you are using MPTS the values for NetBios sessions, names und NCBS are set to a maximum according to the adapter by default.

If you are using APPC connections to NVDM/MVS or to another CC server, you have to increase the link stations defined in the 802.2 section of PROTOCOL.INI by 2+MaxClients.

MaxClients is a keyword in IBMNVDM2.INI that defines the maximum number of clients that can be concurrently processing Installation requests. Supported values are 1 to 100.

MaxRequests is a keyword in IBMNVDM2.INI that defines the maximum number of threads that the NetView DM/2 base uses to process CC client file access requests. Supported values are 1 to 8.

For all of the considerations of this chapter please see also the *IBM NetView Distribution Manager/2 Version 2.1 Installation and Customization Guide*, SH19-6915-05.

Chapter 15. OS/2 CID Utilities

Note: If you are using NetView DM/2, the root of the CID directory structure may be SHAREA. So when following the examples below use D:SHAREAEXEOS2Vxx instead of D:CIDEXEOS2Vxx. Instead of the D: drive remember to use whatever drive you have your CID directory structure on.

For the scenarios in this book, the OS/2 CID utilities are loaded to the directory EXEOS2Vxx, where xx should be substituted with the version number. Be careful to reflect the position of the utilities in your change files, no matter where you put them.

If not mentioned otherwise, the descriptions in this chapter are valid for OS/2 V2.11, OS/2 Warp V3 and OS/2 WARP Connect V3. If there is no specific comment on OS/2 WARP Connect V3 the explanations for OS/2 Warp V3 are also valid for OS/2 WARP Connect V3.

15.1.1 Loading OS/2 CID Utilities to the Code Server

Four programs are shipped with OS/2, which are necessary for successful CID installation of OS/2. These programs are packaged in a bundle file, CID, shipped on diskette 7 of the OS/2 package. The programs are:

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

These files will *not* be installed as part of a normal OS/2 installation, but must be manually unpacked from diskette 7. The bundle file on diskette 7 is called CID. The CID bundle *can* appear on another diskette. For a new OS/2 version 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.

SEINST.EXE calls RSPINST.EXE to do the actual response file driven OS/2 installation. RSPINST.EXE is in a bundle file REQUIRED on OS/2 diskette 7.

SHPIINST.DLL is needed if you want to install printers using the RINSTPRN program. For more information see Chapter 7, "Remote Multiple Printer Support" on page 217. For OS/2 V2.11 SHPIINST.DLL is in the bundle file

REQUIRED on OS/2 V2.11 diskette 7. For OS/2 Warp V3 SHPIINST.DLL is in bundle file BUNDLE on OS/2 Warp V3 diskette 2.

To use the appropriate level of the OS/2 CID Utilities is extremely important. These utilities are version dependent. This is the reason that there are EXE and DLL directories for different OS/2 versions in the CID directory structure.

A sample OS/2 V2.11 response file is provided as SAMPLE.RSP in a bundle file REQUIRED on OS/2 V2.11 diskette 11. For OS/2 Warp V3 the SAMPLE.RSP is in bundle file REQUIRED on diskette 7. The SAMPLE.RSP **must** be modified before you can use it. Please see 3.2.1, "OS/2 Response File" on page 48.

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.

Copy UNPACK commands

For OS/2 V2.11 UNPACK.EXE and UNPACK2.EXE are on diskette 2.

COPY A:UNPACK*. * D:CIDEXEOS2V211

For OS/2 Warp V3 UNPACK.EXE and UNPACK2.EXE are on diskette 2.

COPY A:UNPACK*. * D:CIDEXECONNECT

Refer to the *OS/2 Command Reference* for full details of the OS/2 UNPACK commands.

Loading OS/2 CID utilities can be done as described below or by using 15.1.1.1, "LCU Utility GETOSCID" on page 376.

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 code server you probably will not run GETREXX and therefore the manual steps to unpack/copy them are included below.

OS/2 V2.11 CID Utilities Unpack Example

CD D:CIDEXEOS2V211

The CID and REQUIRED bundles are on OS/2 V2.11 diskette 7.

UNPACK A:CID D:CIDEXEOS2V211

UNPACK A:REQUIRED D:CIDEXEOS2V211 /N:RSPINST.EXE

UNPACK A:REQUIRED D:CIDDLLOS2V211 /N:SHPIINST.DLL

The REQUIRED bundle with SAMPLE.RSP is on OS/2 V2.11 diskette 11.

UNPACK A:REQUIRED D:CIDRSPSOS2V211 /N:SAMPLE.RSP

OS/2 Warp V3 CID Utilities Unpack Example

CD D:CIDEXECONNECT

The CID and REQUIRED bundles are on OS/2 Warp V3 diskette 7. The following assumes that you have the diskettes of Operating System available. If you have the CD ROM version available, change to the CD ROM drive, to the subdirectory OS2IMG and then to the corresponding directory of the diskette that is used, for example DISK_7.

UNPACK A:CID D:CIDEXECONNECT

UNPACK A:REQUIRED D:CIDEXECONNECT /N:RSPINST.EXE

UNPACK A:REQUIRED D:CIDRSPCONNECT /N:SAMPLE.RSP

The BUNDLE bundle with SHPIINST.DLL is on OS/2 Warp V3 diskette 2 which also includes INSCFG32.DLL.

UNPACK A:BUNDLE D:CIDDLLCONNECT /N:INSCFG32.DLL

UNPACK A:BUNDLE D:CIDDLLCONNECT /N:SHPIINST.DLL

And for a successful installation of OS/2 Warp V3 on an HPFS formatted partition, the driver UHPFS.DLL is needed:

COPY A:UHPFS.DLL D:CIDDLLCONNECT

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

1. Install the OS/2 images to the code server first.

Which requires that you have unpacked the correct SEIMAGE.

2. Change to the OS/2 image directory of the version you are interested in.

For OS/2 V2.11 that would be D:\CID\IMGOS2V211

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.

4. Use the /SHOW parameter together with UNPACK to show the content without unpacking anything.

```
D:\CID\XEOS2V211\DISK_1 UNPACK * /SHOW | MORE
```

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

If you are updating the code server using the OS/2 Corrective Service Facility, it will replace the OS/2 CID utility files in **any** location on the code server's hard disk. Be sure to exclude the EXE and DLL directories for other OS/2 versions in the CID directory structure when running OS/2 Corrective Service Facility. For more information about OS/2 Corrective Service Facility refer to Chapter 5, "Maintenance and Service" on page 183.

If you are adding a OS/2 Service Pak and want to CID install it you may need to update the OS/2 CID utilities. Please refer to README on the Service Pak.

15.1.1.1 LCU Utility GETOSCID

GETOSCID assumes that the OS/2 diskette images are already installed on the code server. In order to do that you have to manually unpack SEIMAGE (as described above) and then you can unpack all CID utilities as well, so if you do not have a very special wish to use GETOSCID skip this!

This utility copies the OS/2 CID modules from the OS/2 diskette images that are required to the code server. 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

The NTS/2 GETOSCID did not copy UNPACK2.EXE, which is needed for successful installation of OS/2 V2.1 ('Salmon' package) and later OS/2 versions.

An updated version of GETOSCID.CMD is provided with the sample code CDROM in the SAMPLE directory. This version will work for OS/2 V2.0, OS/2 V2.1 (both 'Blue' Package and 'Salmon' Package), OS/2 V2.11 and OS/2 Warp V3.

GETOSCID Syntax

GETOSCID <source path> <target path>

<source path> Fully qualified source path

The fully qualified path of the OS/2 diskette images

<target path> Fully qualified target path

The name of the subdirectory where the files should be copied

MPTS GETOSCID Example

GETOSCID D:\cid\img\CONNECT D:\cid\exe\CONNECT

15.1.2 SEDISK

SEDISK.EXE is a utility which will create the 'Installation diskette' (DISK_0) and 'Diskette 1' (DISK_1) of OS/2. The diskettes as created have no LAN transport drivers or redirector code. These must be added to the diskette created secondly.

SEDISK requires two formatted diskettes and also requires that the diskette images have previously been installed on the hard disk (using SEIMAGE).

SEDISK Syntax

SEDISK /S:<source path> /T:<target drive> /P:<pcmcia_id#>

/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 are creating boot diskettes for a system with PCMCIA a LAN adapter (an IBM Thinkpad for example), use the /P: option. The PCMCIA.SYS driver (as well as the appropriate socket driver) will be copied over to the boot drive. The pcmcia_id# represents a number associated with the computer desired. Look at the default response file at the keyword PCMCIA to figure out what number to put in here. For example: Specify /P:17 for an IBM Thinkpad 750.

SEDISK Example

D:\cid\exe\CONNECT\SEDISK /S:D:\cid\img\CONNECT /T:A:

The command above will create the two OS/2 WARP Connect V3 bootable diskettes from files in the D:\CIDIMGCONNECT subdirectory. If you wish to create diskettes for a different version than OS/2 WARP Connect V3 change the paths accordingly.

Note: If the client machine has some special requirements, like PCMCIA drivers, you have to add these manually to the LAN Transport System diskette's CONFIG.SYS file. Please see Appendix I, "Hardware and Software Dependencies" on page 571.

Chapter 16. Loading Product Images to Code Server

Note: If you are using NetView DM/2, the root of the CID directory structure may be SHAREA. So when following the examples below use D:SHAREA... instead of D:CID... Instead of the D: drive remember to use whatever drive you have your CID directory structure on.

For those products below where it is applicable there also is a description on how to load the sample response files to the code server. It is a good idea to do this at the same time as you load the product images, when you have the product's diskettes/CD-ROM available.

16.1.1 Loading OS/2 Diskette Images with SEIMAGE

SEIMAGE.EXE is a utility to automate the creation of a subdirectory structure on the CID code server, for use by the installation process. SEIMAGE copies all OS/2 diskettes to a specified target directory. 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.

For OS/2 Warp V3 after the OS/2 diskettes are copied a choice is given to create a tree structure for Windows. If the user answers with a Y(es), SEIMAGE prompts for a directory name. This name will be appended to the current target directory (as specified by /T). The user is then prompted to feed the Windows diskettes. Since OS/2 Warp V3 can be installed on top of Windows 3.1 or 3.11 or Windows for Workgroups 3.1 or 3.11 the choice is given to create a tree structure for another Windows package.

Note: If Windows should be supported under OS/2 Warp V3, DOS and Windows have to be installed on the client machine before the OS/2 installation program is run. The Windows directories in the CID structure are only used so that the OS/2 installation program can copy some unmodified Windows files during the OS/2 Warp V3 installation.

SEIMAGE Syntax

```
SEIMAGE /S:<drive> /T:<target path>
```

/S: Source drive

This is the diskette drive from which the diskettes will be loaded.

/T: Fully qualified target path

This is the fully qualified target directory name. This directory will be shared and become the source path for installation by RSPINST.EXE.

SEIMAGE Example for OS/2 Warp V3

```
D:\cid\exe\CONNECT\SEIMAGE /S:A: /T:D:\cid\img\CONNECT
```

The command above will install from diskettes in the A: drive to a directory structure under D:\cidimgCONNECT. The following figure shows the directory structure which will be created.

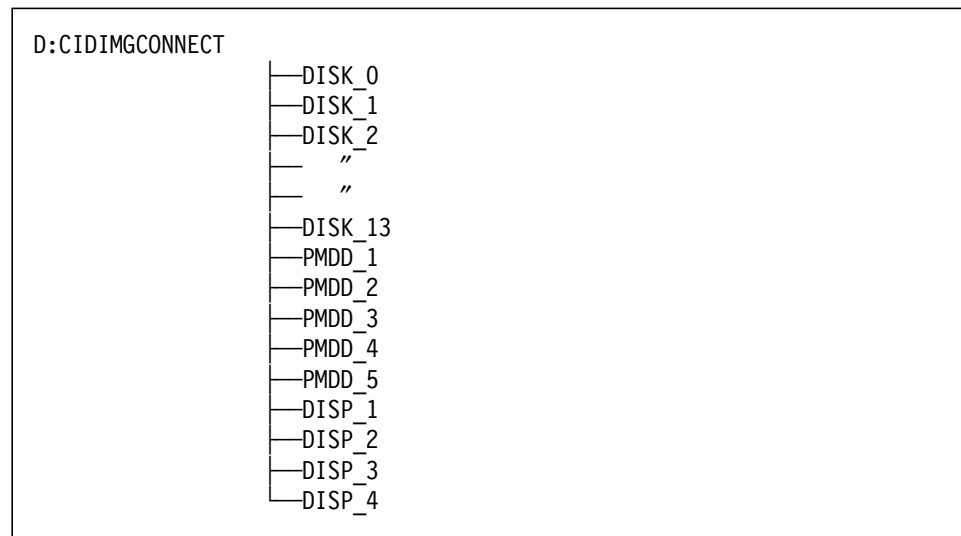


Figure 91. OS/2 Warp V3 SEIMAGE Directory Structure

16.1.1.1 Loading OS/2 from CD-ROM

The OS/2 CD-ROM has the necessary directory structure, so the easiest way is to make an XCOPY of it. Assuming that the CD-ROM is connected as E: the command would be:

— OS/2 WARP Connect V3 XCOPY from CD-ROM Example —

```
XCOPY E:\OS2IMG\*. * D:\cid\img\CONNECT /S /E /V
```

16.1.1.2 Loading OS/2 Warp V3 Images to an OS/2 V2.x Based Code Server

OS/2 Warp V3's Install Diskette and Diskette 1 are in the 'normal' diskette format. Subsequent diskettes are formatted with an extended diskette format. The XDF is a higher capacity diskette image format, which reduces the number of diskettes and the time needed for the OS/2 installation.

To enable OS/2 Warp V3 SEIMAGE to put code on a code server running OS/2 V2.x either of the following methods can be used:

1. Update the OS/2 diskette driver on the code server.
 - a. Copy XDFCOPY.EXE from the OS/2 Warp V3 Install Diskette to the OS2 directory.
 - b. Copy XDFLOPPY.FLT from the OS/2 Warp V3 Diskette 1 to the OS2 directory.
 - c. Add the following line to the CONFIG.SYS:

BASEDEV=XDFLOPPY.FLT

For an ISA-bus system:
 - d. Rename IBM1FLPY.ADD to IBM1FLPY.OLD in the OS2 directory.
 - e. Copy IBM1FLPY.ADD from OS/2 Warp V3 Diskette 1 to the OS2 directory.
 - f. For a Micro Channel system:
 - f. Rename IBM2FLPY.ADD to IBM2FLPY.OLD in the OS2 directory.
 - g. Copy IBM2FLPY.ADD from OS/2 Warp V3 Diskette 1 to the OS2 directory.
2. Boot OS/2 Warp V3 from diskettes.
 - a. Boot on OS/2 Warp V3 Install Diskette and when requested to change to Diskette 1 do so.

- b. On the "Installing Operating System/2" panel press the F3 key to get an OS/2 command prompt.
- c. Assuming that the steps described in 15.1.1, "Loading OS/2 CID Utilities to the Code Server" on page 373 have been previously performed execute SEIMAGE as described above from the directory with the OS/2 Warp V3 versions of UNPACK*.EXE and CID utilities.

16.1.1.3 Unpacking the Sample Response file SAMPLE.RSP

For OS/2 the SAMPLE.RSP is packed in a bundle file REQUIRED. This bundle file may be placed on a different diskette for a different OS/2 version. And the sample response file is version dependent since new keywords have been added and old keywords may be set to different values.

For your convenience we have provided a REXX file GETSAMP.CMD on the sample code CDROM to help you unpack the SAMPLE.RSP file.

GETSAMP Syntax

GETSAMP <source path> <target path>

<source path> Fully qualified source path

The fully qualified path of the OS/2 diskette images.

<target path> Fully qualified target path

The name of the subdirectory where the SAMPLE.RSP file should be copied.

GETSAMP for OS/2 Warp V3 Example

D:\cid\img\sample\GETSAMP D:\cid\img\CONNECT D:\cid\rsp\CONNECT

16.1.2 Loading LAN Transport System Diskette Image(s) with LAPSDISK

LAPSDISK is a CID utility for copying the image of NTS/2 LAPS diskette or the MPTS diskettes (1 and 2) to a code server. LAPSDISK requires the OS/2 XCOPY command.

LAPSDISK Syntax

LAPSDISK <source path> <target path>

<source path> Fully qualified source path

For NTS/2 LAPS this specifies source drive of the LAPS diskette or the source drive and path of a LAPS diskette image.

For MPTS this specifies the source drive for the MPTS diskettes.

<target path> Fully qualified target path

Specifies LAPS target drive and path on the code server.

LAPSDISK Example

```
A:\LAPSDISK A:\ D:\cid\img\laps
```

The command above will copy the image of the LAN Adapter and Protocol Support diskette(s) from diskette drive A: to directory D:CIDIMGLAPS.

NTS/2 LAPS Update for OS/2

The latest version of LAPSCID.DLL is required for successful redirected installation of OS/2. Reference APAR IC06187. The new LAPSCID.DLL is included in NTS/2 CSD WRx7040 (or later).

16.1.2.1 Copying of Sample LAN Transport Response Files to the Code Server

Sample NTS/2 LAPS response files are provided on the NTS/2 Utilities diskette in the SAMPLE directory:

Copying of LAPS Sample Response Files Example

```
XCOPY A:\sample\laps*.rsp D:\cid\rsp\laps
```

Sample MPTS response files are provided on the MPTS Utilities diskette in the SAMPLE directory. They are packed together in SAMPLE.ZIP:

Unpacking of MPTS Sample Response Files Example

```
PKUNZIP2 A:\sample\sample D:\cid\rsp\mpts
```

16.1.3 Loading Communications Manager/2 Diskette Images

CMIMAGE is a CID utility for copying the images of CM/2 diskettes to a code server. CMIMAGE requires you to answer a couple of questions.

CMIMAGE Syntax

```
CMIMAGE /S:<drive> /T:<target path>
```

/S: **Source drive**

This is the diskette drive from which the CM/2 V1.11 diskettes will be loaded.

/T: **Fully qualified target path**

Specifies CM/2 V1.11 target drive and path on the code server.

CMIMAGE Example

```
A:\CMIMAGE A:\ D:\cid\img\cm2111
```

First you will get some information and a question if you want to continue to transfer files. Reply with:

1 (1=continue is default, 0=cancel)

Then you will get a message, CMI0011, that the directory could not be created because it already exists and that files might be replaced. **Reply** with:

1 (1=continue, 0=cancel is default)

The command above will copy the image of the CM/2 V1.11 diskettes from diskette drive A: to directory D:\CID\IMGC\CM2111 and log to CMIMAGE.LOG in the same directory.

16.1.3.1 Loading Communications Manager/2 from CD-ROM

The easiest way to load CM/2 is to XCOPY the CM2 directory from the CD-ROM. The paths are defined as described for CMIMAGE.

Note: The CM2UNZIPPED subdirectory should not be copied.

Assuming that the CD-ROM is connected as E: the command would be:

— XCOPY from CD-ROM Example —

```
XCOPY E:\CM2 D:\cid\img\cm2111 /V/S/E
```

16.1.3.2 CM/2 V1.11 Distributed Feature Code Server

If you want the code server to act as a CM/2 Distributed Feature server run the CMIMAGE /U from the CM/2 V1.1 image directory created above to have a directory with all files unpacked in it.

— CMIMAGE /U Example —

```
D:\cid\img\cm2111\CMIMAGE D:\cid\img\cm2111 D:\cid\img\cm2111df /U
```

16.1.3.3 Copying CM/2 V1.11 Sample Response Files

CM/2 V1.11 sample response files are delivered on one of the diskettes that comes with CM/2 V1.11. On the CM/2 V1.11 CD-ROM version they can be found in the RESPONSE directory.

— Copying of CM/2 V1.11 Sample Response Files Example —

```
XCOPY A:*.rsp D:\cid\rsp\cm2111
```

16.1.4 Loading DB2/2 Diskette Images

There is no special CID utility to load the DB2/2 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.

— Warning —

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 ... for example DDCS/2)
- We tested the first three products in our lab. The directory structure for DB2/2 V2.11 in the CIDIMG directory is as follows:
 - there is a parent directory DB2V211
 - with subdirectories for the different products
 - DB2SRVR
 - DB2SU
 - DB2CAE2

Syntax for copying of DB2/2 diskette

XCOPY <source drive> <target path> /S /E

If your source medium is a CD-ROM instead of diskettes:

- 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
 - (other languages)
- XCOPY this directory to your code server. For example:


```
XCOPY <CD-ROM drive>:EN*.* D:CIDIMGDB2V211 /S /E
```

16.1.5 Loading LAN Server Diskette Images

When loading LAN Server V4.0 or later diskette images to the code server the normal installation program, LANINST, is used. In the examples below LAN Server V4.0 was used. The examples are valid for the LAN Server V5.0 too because there is no change in the functions described below.

1. Insert the LAN Server Installation Diskette 1 into the code server's diskette drive.
2. Type the installation command:

LAN Server Installation Command

A:\LANINST

3. When the IBM logo is displayed, select **OK**.
4. For LAN Server V4.0 select Tailored on the "Easy or Tailored Installation/Configuration" window.

The Installation Task window is displayed (Figure 92).

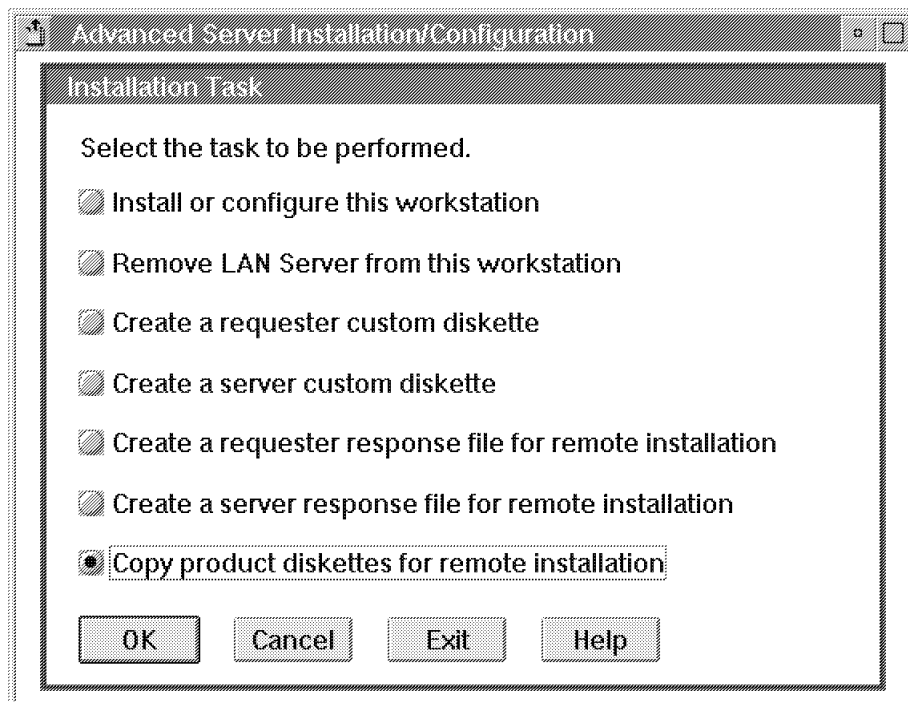


Figure 92. LAN Server V4.0 Installation Task Window

5. Select **Copy product diskettes for remote installation** in order to copy the product diskettes to the code server. Select **OK**. The Copy Product Diskettes window is displayed (Figure 93 on page 388).

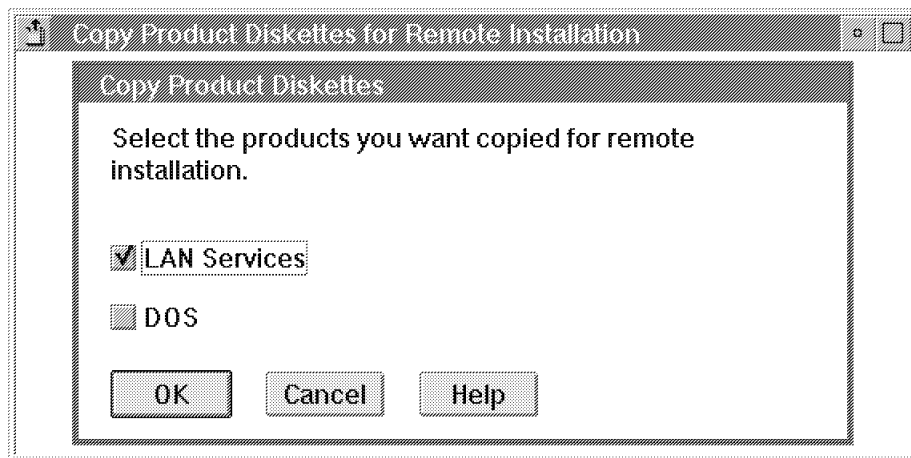


Figure 93. LAN Server V4.0 Copy Product Diskettes Window

6. Select LAN Services or LAN Services and DOS. DOS would be selected if the code server is going to remotely install a LAN Server supporting DOS RIPL. DOS images can be added any time, so if you wish you can add DOS later. Select **OK**.
7. If DOS was selected in the previous step, the following versions of DOS are supported by LAN Server V3.01 and LAN Server V4.0:
 - IBM DOS Version 6.3, 6.1, 5.0 and 3.3
 - Microsoft** DOS Version 6.0, 5.0 and 3.3

In addition LAN Server V4.0 also supports Microsoft DOS 6.2.

Note

If you choose to install DOS 5 you will need to create a DOS startup diskette prior to this step. A DOS startup diskette can be created simply by installing DOS onto diskette(s).

8. On the Remote Install Subdirectory window enter a target path to where the IBM Operating System/2 Local Area Network Server Version product diskettes are to be copied. If you are installing LAN Server V4.0 Entry version, the directory path is:

D:\CID\IMG\LSE40

Or if you are installing LAN Server V4.0 Advanced version, the directory path is:

D:\CID\IMG\LSA40

Select **OK** to start the copy process.

Note

If you specify a path that does not currently exist, it will be created for you. If an existing path is specified, all existing subdirectories beginning with the prefix IBM (except IBMDOSxx) within that subdirectory path, and their contents will be removed.

9. Insert product diskettes as prompted.

When the product diskettes have been successfully copied to the remote installation subdirectories on the code server's fixed disk, select **OK** to continue. The Installation Task window is displayed again (see Figure 92 on page 387). Select **Exit** here to end the LAN Server installation program.

If LAN Server V5.0 Advanced was installed the LAN Server V5.0 directory structure would look as shown below.

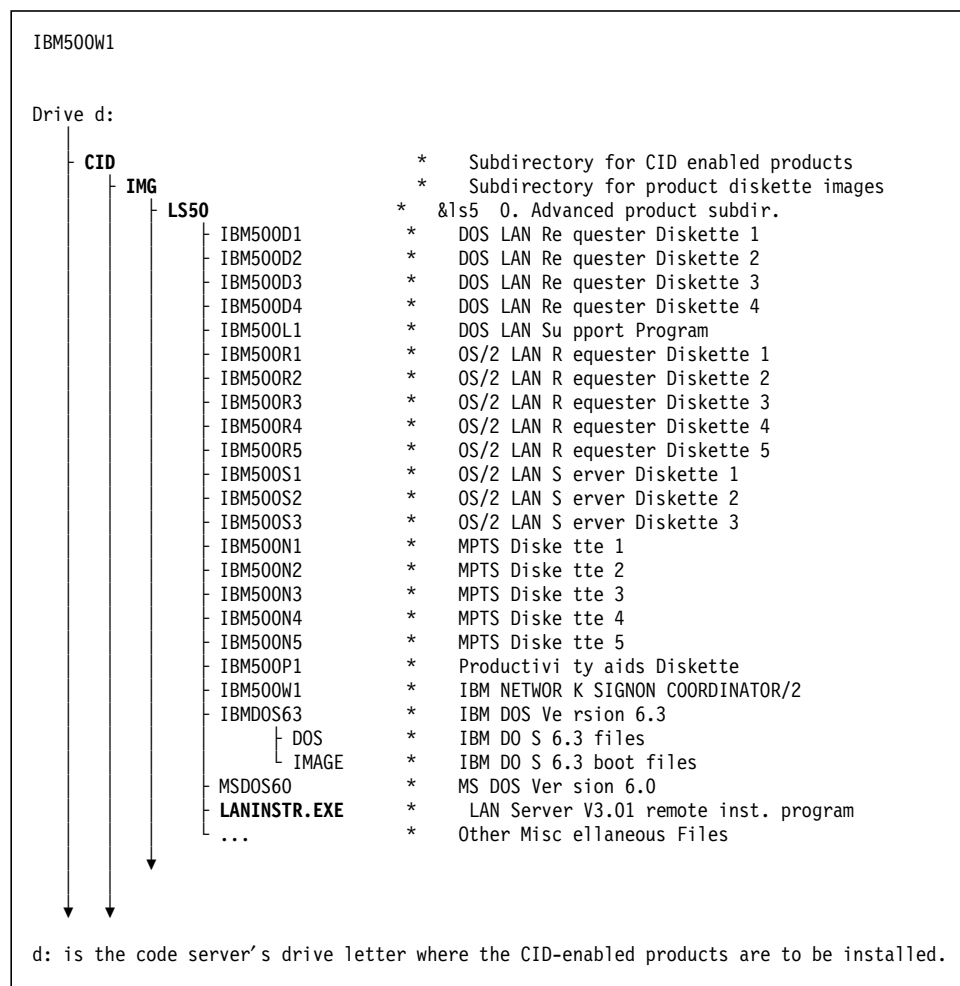


Figure 94. LAN Server V5.0 CID Subdirectory Structure

For the complete CID directory structure please see Chapter 2, "Recommended CID Directory Structure" on page 39. LAN Server V5.0. LANINST will not copy the NTS/2 or MPTS diskettes into the structure.

16.1.5.1 Loading LAN Server V5.0 from CD-ROM

The CD-ROM has the required directory structure, so the easiest way is to use XCOPY. Assuming that the CD-ROM is connected as E: the command would be:

XCOPY for LAN Server Advanced V5.0 from CD-ROM

```
XCOPY E:\IBMLSA D:\cid\img\lsa40 /S /E /V
```

Note: The XCOPY above is valid if it is an Advanced version of LAN Server. If it is the Entry version the directory on the CD-ROM is IBMLSE and the target on the code server is CIDIMGLSE50.

16.1.6 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:

```
NVDMCOPY /S:A:\ /T:D:\SHAREA\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, 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 V2.0 diskette 2 into drive A:

```
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.

IMPORTANT

There is no need to apply the CSD if your NetView DM/2 diskettes are already at the latest SYSLEVEL. As of May 1996, the latest CSD was XR20466A, equivalent to SYSLEVEL XR00002. The FixPak called XREFP01 changing the SYSLEVEL to XR00003 is also recommended.

16.1.7 Loading TCP/IP Diskette Images

There is no special CID command to copy the TCP/IP V2.0 and the TCP/IP V3.0 diskettes to the code server. The description given here is based on TCP/IP V2.0 but it is also valid for TCP/IP V3.0. For TCP/IP V3.0 you can easily create the diskettes from the WARP Connect CD using the LOADDSKF utility. You can find the disk images in the directory \IMAGES\TCPIP on the OS/2 WARP Connect V3 CD .

Each diskette in one of the TCP/IP V2.0 Kits contains a file *name_n.ZIP*. Where name is the abbreviated name of the kit, or a component in the kit, and *n* indicates the number of the diskette within the kit.

Table 14. TCP/IP V2.0 Abbreviated Names. Abbreviated Names of the TCP/IP V2.0 Kits

Kit	Abbreviated Name
Base	BASE
Network File Server	NFS
DOS Box Access	DBOX
Extended Networking	XNT
Programmer's Toolkit	PGMG
X Window System Client Runtime Services	XCLI
X Window System Client Programmer's Toolkit	XCPR
OSF/Motif Runtime Services	MOTIF
OSF/Motif Programmer's Toolkit	MTPR
X Window System Server	PMX
Domain Name Server	DNS

Each kit also needs a *name.EXE* to be copied from the first diskette in the kit to the D:\cidimgtcip20 directory.

For all TCP/IP V2.0 Kits you want to install repeat the following for each diskette:

— **XCOPY of TCP/IP V2.0 Diskette Example** —

```
XCOPY A:\*.ZIP D:\cid\img\tcpip20
```

The command above will copy the *.ZIP file(s) from the TCP/IP V2.0 diskette to directory D:CIDIMGTCP20.

From the **first** diskette of all TCP/IP V2.0 Kits repeat the following:

— **COPY of EXE-file from TCP/IP V2.0 Diskette Example** —

```
COPY A:\nameXT.EXE D:\cid\img\tcpip20
```

From the first diskette of the Base Kit please copy the following files:

— **COPY from TCP/IP V2.0 Base Kit Example** —

```
COPY A:\DEFAULT.RSP D:\cid\rsp\tcpip20  
COPY A:\UNZIP.DLL D:\cid\img\tcpip20  
COPY A:\TCPINST.EXE D:\cid\img\tcpip20  
COPY A:\TCPINST2.EXE D:\cid\img\tcpip20  
COPY A:\TCPINST.HLP D:\cid\img\tcpip20
```

Applying the CSDs

There is also no utility shipped with the CSDs to copy the CSD diskettes to the code server. You can use the XCOPY command. The CSDs are copied to the same subdirectory as the base product. They will not overwrite or corrupt the files of the base product. During an install both base product and CSD will be installed in one process, so that there is no need to keep the CSD separately and apply it in an extra step.

— **XCOPY of TCP/IP V2.0 CSD Diskettes Example** —

```
XCOPY A: D:\cid\img\tcpip20
```

16.1.8 Loading LAN CID Utility Files

You can use either NTS/2 or MPTS LAN CID Utility. If you have access to MPTS, which is shipped with LAN Server V4.0 we recommend that you use these files.

16.1.8.1 Loading NTS/2 LCU Files

There is no utility shipped with NTS/2 to transfer the LCU directory of the Utilities diskette to the code server.

The code server administrator will copy all the LCU files into the LCU image directory.

COPY example

```
XCOPY A:\lcu D:\cid\img\lcu
```

16.1.8.2 Loading MPTS LCU Files

On the MPTS Utilities diskette the LCU files are packed together in the LCULCU.ZIP.

The code server administrator will unzip the LCU.ZIP file into the LCU image directory. PKUNZIP2.EXE can be found on the first MPTS diskette and can be copied from there to any suitable directory in the current PATH.

UNZIP example

```
PKUNZIP2 A:\lcu\lcu.zip D:\cid\img\lcu
```

16.1.9 Loading SRVIFS Files

You can use either NTS/2 or MPTS SRVIFS. If you have access to MPTS, which is shipped with LAN Server V4.0 we recommend that you use these files. But take care to use the same version as for the LAN CID Utility.

16.1.9.1 Loading NTS/2 SRVIFS Files

There is no utility shipped with NTS/2 to transfer the SRVIFS directory of the Utilities diskette to the code server.

The code server administrator will copy all the SRVIFS files into the SRVIFS image directory.

COPY example

```
XCOPY A:\srvifs D:\cid\img\srvifs
```

In the SAMPLE directory there is a sample SERVICE.INI file and a sample authorizations list file SERVICE.LST.

Copying of SRVIFS sample files

```
XCOPY A:\sample\service.*  
D:\cid\img\srvifs
```

16.1.9.2 Loading MPTS SRVIFS Files

On the MPTS Utilities diskette the SRVIFS files are packed together in the SRVIFSSRVIFS.ZIP.

The code server administrator will unzip the SRVIFS.ZIP file into the SRVIFS image directory. PKUNZIP2.EXE can be found on the first MPTS diskette and can be copied from there to any suitable directory in the current PATH.

UNZIP example

```
PKUNZIP2 A:\srvifs\srvifs.zip D:\cid\img\srvifs
```

In the SAMPLESAMPLE.ZIP there is a sample SERVICE.INI file and a sample authorizations list file SERVICE.LST.

SRVIFS sample files example

```
PKUNZIP2 A:\sample\sample.zip D:\cid\img\srvifs SERVICE.*
```

16.1.10 Loading NetWare Requester Files

As the NetWare requester is not yet CID-enabled, there is no utility shipped with Novell NetWare to transfer the NetWare requester files to the code server.

The code server administrator will copy all the NetWare requester files from an installed workstation into the NetWare image directory. This is necessary because we will only copy the files to a client workstation and not execute the NetWare installation program. On the diskettes, the NetWare files are compressed. As the code server administrator is advised to execute the

necessary steps to prepare a code server using Novell NetWare on a workstation running OS/2 V2.x or higher and NetWare requester, the following example shows what to do, assuming that the NetWare requester is installed on C:.

COPY example

```
XCOPY C:\NetWare D:\cid\img\netware
```

Additional procedures are needed to install NetWare requester on a client station. They can be found on the sample code CDROM supplied with this book and should be copied to the D:CIDIMGSAMPLENetWare subdirectory.

Chapter 17. LAN CID Utility

Not all utilities are described here; some have been previously described in the context in which they were used.

These utilities are available with NTS/2 (bought as a stand-alone product or shipped with LAN Server V3.0). The currently (May 1996) available NTS/2 CSD level is WRx7060.

With LAN Server V4.0 and higher NTS/2 is no longer shipped, but MPTS is shipped instead. Updated versions of the LAN CID Utility are provided with MPTS.

The text below assumes that the LAN CID Utility is already loaded on the code server, as described in 16.1.8, "Loading LAN CID Utility Files" on page 394.

17.1.1 GETBOOT

In order to complete the installation process, LCU must be able to reboot the client workstation when appropriate. To do this, it 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.

GETBOOT is a utility shipped on the NTS/2 Utilities diskette and on the MPTS diskette 3. GETBOOT unpacks SETBOOT.EXE and XCOPY.EXE files from the OS/2 diskette images to the code server executable directory.

The currently available version of GETBOOT.CMD from the NTS/2 Utility diskette does not work with OS/2 V2.1, OS/2 V2.11 or OS/2 Warp V3.

On all of the first OS/2 Warp V3 and OS/2 WARP Connect V3 diskettes there will be a README.CID, which includes an updated and working version of GETBOOT.CMD. The version of this utility shipped with MPTS also works.

GETBOOT Syntax

GETBOOT <source path> <target path>

<source path> Fully qualified source path

Fully qualified path of the OS/2 diskette images.

<target path> Fully qualified target path

Fully qualified path of the subdirectory where SETBOOT command should be copied.

GETBOOT for OS/2 WARP Connect V3 Example

D:\cid\sample\GETBOOT

D:\cid\img\CONNECT D:\cid\exe\CONNECT

17.1.2 GETREXX

REXX is required on the code server to run the LCU REXX command files used to install the requested products. Since the client workstation accesses the LCU command files via a redirected drive, it makes sense to access the required files for REXX from that 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 images to the code server executable directory.

The currently available version of GETREXX.CMD from the NTS/2 Utility diskette does not work with OS/2 V2.1, OS/2 V2.11 or OS/2 Warp V3.

The updated GETREXX.CMD delivered with MPTS LCU also copies OSO001.MSG and INSCFG32.DLL to the target path.

On all of the first OS/2 Warp V3 and OS/2 WARP Connect V3 diskettes there will be a README.CID, which includes an updated and working version of GETREXX.CMD.

This GETREXX.CMD will unpack all REXX bundle files, which currently includes:

- REX.MSG
- REXH.MSG

- REXX.DLL
- REXXAPI.DLL
- REXXINIT.DLL
- REXXTRY.CMD
- REXXUTIL.DLL
- RXQUEUE.EXE
- RXSUBCOM.EXE (for OS/2 Warp V3 only)

In addition it will unpack/copy:

- SHPIINST.DLL
- UHPFS.DLL

For future OS/2 versions please look for README.CID files with the latest information on any of the first couple of diskettes. An easy way to find README files is to search the OS/2 images on the code server. For example for OS/2 Warp V3 do an

```
D:\CID\IMG\CONNECTDIR READ*. * /S
```

This will find for example README.CID and README.INS. Both should be read before any installation is done.

GETREXX Syntax

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

<source path> Fully qualified source path

The fully qualified path of the OS/2 diskette images.

<target path> Fully qualified target path

The name of the subdirectory where the REXX files should be copied.

GETREXX Example

```
D:\cid\img\sample\GETREXX
D:\cid\img\CONNECT D:\cid\d11\CONNECT
```

Earlier experiences on code servers running OS/2 V2.1 or OS/2 V2.11, with 8MB memory or less, are 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.

17.2 Quick Reference

Product	Program	Syntax of Parameters
OS/2	SEIMAGE	/S:<source drive> 1 /T:<target path> 1
	SEDISK	/S:<source path> 1 /T:<target drive> 1 /P:<pcmcia#> 2 :
	SEMAINT	/S:<source path> 1 /T:<target path> 1 /B:<boot drive> 1 /L1:<path><log file name> 1 /S2:<service pak path>: /P:<pcmcia#> 2 :
	SEINST	/B:<target boot drive> 1 /T:<boot directory> 2 /R:<path><response file> 1 /S:<source path> 1 /L1:<path><log file name> 1

LCU	CID utilities	
	LAPS or MPTS	/S:<source path> 3 /T:<target path> 2 /R:<path><response file> 1 /G:<search path> 2 /TU:<config.sys path> 2 /E:<PREP MAINT PROD> 2 /L1:<path><log file name> 2
	LAPSDISK	<source path> 1 <target path> 1
	LAPSRSP	<source path> 1 <target path> 1 /T:<drive> 2 /I:<PRODUCT ADDITIONAL> 2 /C:<cfg_path_name> 2 /U:<OLD SAME NEW> 2 /M:<MPTS configuration file> 6 /N:<names list file> 6 /B:<broadcast list file> 6 /V:<resolv list file> 6
	THINLAPS	<source path> 1 <target> 1 <NIF file name> 1 /P:<path><file name> 1
	LAPSDDEL	No parameters
	THINSRV	/S:<source path> 1 /R:<path><response file> 1 /T:<target path> 2 /TU:<config.sys path> 2 /U:<authlist file name> 2 /L1:<log file name> 2
	THINIFS	/S:<source path> 1 /T:<target path> 2 /TU:<config.sys path> 2 /A:<0 1> 2 /L1:<path><log file name> 2 /REQ:<redirector name> 1 /W 2 /SRV:<server name> 1 /D:<drive> 1 /NS:<2-9> 2
	IFSDEL	/T:<target path> 2 /TU:<config.sys path> 2 /SD 1
	SERVICE.EXE	/INI /QU /F /ST /AU 5
	CASINSTL	/TU:<boot drive> 1 /CMD:<cmd file path> 1 /D <default cmd file name> 2 /L1:<path><log file name> 2 /L2:<path><log file name> 2 /PL:<libpath,dpath> 2 /PA:<path> 2 /O 2 /PD:<boot disk path> 6 /REQ:<LCU client name> 6
	CASAGENT	<CMD file path> 4 /L1:<path><log file name> 4 /D 4 /REQ:<LCU client name> 6
	CASDELET	/TU:<boot drive> 1 /PL:<libpath,dpath> 1 /L1:<path><logfile name> 2
	GETBOOT	<source path> 1 <target_path> 1
	GETFIX	<source path> 1 <target_path> 1
	GETREXX	<source path> 1 <target_path> 1
	GETOSCID	<source path> 1 <target_path> 1
Note: 1 Required parameter 2 Optional parameter 3 Required parameter for LAPS unattended install 4 Parameters are supplied by CASINSTL and put in STARTUP.CMD file 5 Parameters are used to start, stop, force stop, query status and update authorization list of code server. 6 Optional parameter (MPTS)		

Table 15. Remote Installation of OS/2 Command Quick Reference

Chapter 18. Automated Setup with CASSETUP

A Presentation Manager based program called CASSETUP is provided on the utilities diskette of MPTS (Diskette 3 of MPTS). It assists the administrator in preparing the code server. CASSETUP resides in the APPLETs subdirectory of the utilities diskette. It is described in detail in the *LAN CID Utility Guide*, S10H-9742, in Appendix I. Support of OS/2 Warp V3 is provided with the Service Pak WRx8150 of MPTS. If you want to use CASSETUP on a system running OS/2 Warp V3 or distributing OS/2 Warp V3 this CSD is recommended. Please refer to the CASSETUP.INF file for more and updated information of the product.

Older Version of CASSETUP with NTS/2

Coming with NTS/2, there is an older version of CASSETUP that can be found in the APPLETs subdirectory of diskette 3. This version is not GUI based and has a lot of restrictions that are detailed in the README.UTL. If you want to install OS/2 Warp V3 this version will not work. Therefore, it is recommended to use the MPTS version for installation of OS/2 Warp V3 and related levels of system software.

18.1 Functions of CASSETUP

The Setup Utility provides the following functions to assist administrators in preparing for redirected installation:

1. A Presentation Manager Interface for the installation and removal of Redirected Installation Support. Redirected Installation Support includes the LAN CID Utility (LCU) and the Service Installable File System (SRVIFS).
2. Initial configuration and reconfiguration of SRVIFS.
3. Installation and removal of diskette images for:
 - OS/2 V2.1, OS/2 V2.11, OS/2 Warp V3
 - IBM Multi-Protocol Transport Services
 - IBM LAN Server V4.0
 - Other applications, with the usage of profiles. See the online documentation for more information about these profiles.

4. Creating CASPREP input files that allow clients to install any combination of product images that were either installed on or registered with the code server through CASSETUP.
5. Creating boot diskettes for use by clients to initialize and process redirected installation sessions.

If you want to use CASSETUP thoroughly, it is useful to be familiar with the structure and concept of CASPREP. This is detailed in the *LAN CID Utility Guide*, S10H-9742. It is also helpful to know about the structure of the LCU command files. The corresponding chapters of this book are useful with this task.

18.2 Requirements

1. OS/2
CASSETUP runs on OS/2 V.2.1 or higher.
2. MPTS LAPS
MPTS LAPS must be installed and configured for NetBIOS before you can use CASSETUP to install redirected installation support. Refer to the *MPTS Configuration Guide*, S10H-9693 for information on installing and configuring LAPS.
3. REXX
Redirected installation support requires REXX. Therefore, CASSETUP requires REXX to be installed on the workstation before you can install it.

Chapter 19. Migration and How to Add New Products

This chapter discusses the migration of existing code servers to new products or new levels of already used products and the required changes to the software distribution manager implementation used.

19.1 Code Server Migration

This section discusses changes that can be done to an existing code server.

If you have a code server running, it is very easy to change it to another version, for example the structure that is described in this book.

The code server you have running might differ from the way described here in one or more of the following parts:

- The version of the LAN CID Utility used
- The products that are distributed
- The structure of the LCU command files used to install a client workstation

If you are using a version of LAN CID Utility other than the one used in this book, and it is running without problems, there is no need to migrate it. If you want to migrate it, run the code server installation as described in Chapter 11, "Manual Setup of LAN CID Utility" on page 293 again.

If you want to add a more recent version of a product than the one that is already on the code server, you have to be aware of some version-specific files that might have the same names. For most products, it is enough to create a new directory under D:CIDIMG assuming that the CID directory structure is located on your D: drive. Then use the product's way to transfer the image to this directory. Please refer to Chapter 16, "Loading Product Images to Code Server" on page 379 for a detailed description. If you want a code server to be able to install different OS/2 versions, there have to be different D:CIDEXE and D:CIDDLL subdirectories. The CONFIG.SYS of the client boot diskettes have to point to the correct subdirectory of the version you want to install. Additionally, the boot diskettes have to be recreated for the new version because the boot diskettes must be of the same OS/2 version as the one that is to be installed.

If you want to use LCU command files from different sources, there is no problem to use them at the same time. You have to ensure that they are all adapted to the directory structure you are actually using. To ensure an effective administration of the installs the number of different command files used should be limited.

19.2 Migrating a Code Server from NTS/2 LCU to MPTS LCU

The things that might be necessary to change are the CID directory structure and the LCU command files. If the CID directory from the redbooks preceding this one is used it does not need to be really changed only expanded with new directories.

MPTS LCU command files offer a couple of new interesting features and if these will be utilized the command files need to be updated.

Please see "Migrating from previous LAN CID Utility Command Files and Directory Structures" in the *LAN CID Utility Guide*, S10H-9742.

19.3 Migrating a Code Server from LCU to NetView DM/2

This section gives you an advisory for a change in the software distribution manager product that is used.

The change of the software distribution manager product from LCU to NetView DM/2 does not affect the following parts of the CID installations:

1. The CID directory structure as described in Chapter 2, "Recommended CID Directory Structure" on page 39.
2. The response files.

It does, however, affect the way an install is organized:

LCU uses the LCU command files to manage the installation of the client workstation. NetView DM/2 uses change files to keep the information about a product install and uses its own commands to install these change files on a client machine. Please refer to 4.6, "NetView DM/2 Change Control Files" on page 171 and to the NetView DM/2 manuals, especially the *IBM NetView Distribution Manager/2 Version 2.1 Installation and Customization Guide*, SH19-5048-02, for detailed information on that. When changing to NetView DM/2, you will need to create change files for all products that are to be installed. Please refer to 4.6, "NetView DM/2 Change Control Files" on

page 171 for information on how to create these change files. You will no longer use the LCU command files.

As a lot more DASD space is needed for NetView DM/2 than for LCU (the product subdirectory needs at least 15MB, and the database that is created during the install needs additional space) it is a good idea to use a new system rather than migrating the existing space that might run into space problems. The new system can copy the complete CID directory structure from the existing server as soon as it is connected to the LAN and then migrate following the guidelines of this chapter.

19.4 Adding New Products

To add other products to the code server, you will have to follow these steps:

1. Load the product diskette images to the code server.
 - Review the product's README file.
 - Ensure that you have the required disk space for all product images.
2. Create the appropriate directory structure. Remember to add directories for the log files and for the response files.

Refer to the product documentation if there is a utility supplied or a recommended way to load the images. See Chapter 16, "Loading Product Images to Code Server" on page 379 for more details on all the products covered in this book.

3. Create the response files and add the necessary keyword parameters.

Refer to the product documentation if there is a sample response file supplied or a utility to create response files. See Chapter 3, "Response Files" on page 47 for details about the response file usage of the products covered in this book.

4. For NetView DM/2 code servers:

Create appropriate change files. See 4.6, "NetView DM/2 Change Control Files" on page 171 for information on how to create change files. Refer to the product documentation if there is a sample change file profile supplied. If not, check the product documentation for a detailed description of the invocation syntax of the install program. Check if you need to reboot the client workstation after the install. If you are adding a CID enabled product the install program should return an appropriate return code and force a reboot if it is required. If you are adding a product that is not CID enabled, you might have to force the reboot. This could be done by adding *PhaseEnd=Yes* in the change file profile if the

product is installed in a coreq group. The reboot will then take place directly after the product install regardless of other products that might follow in the coreq group. If there is any product included in the coreq group that automatically forces a reboot after the coreq group is installed completely, there is no need to add anything for the new product. You might also issue an **ACTIVATE** to force a reboot.

5. For LCU code servers:

Create appropriate LCU command files. The product definition part of the LCU command file has to have a definition for the added product. Check if there is a sample LCU command file included. If not, check the product documentation for a detailed description of the invocation syntax of the install program. Do not forget to increase the number of install programs by one. Add the product to the installation queues. See 4.4, "LCU Command File" on page 143 for a detailed description of the LCU command file. Check if you need to reboot the client workstation after the install. If you are adding a CID-enabled product the install program should return an appropriate return code and force a reboot if it is required. If you are adding a product that is not CID-enabled, you have to force the reboot by adding *RebootAndGotoState(x)* after the *RunInstall* statement for the product.

Part 4. CID Enabling of Applications

This part provides information for application developers who wish to enable their products for a CID environment.

Due to the availability of an IBM publication on the subject, we decided not to replicate that info here. Please order the publication *CID Enablement Guidelines* (S10H-9666-01).

The following chapter on Software Installer is new to this edition of the book.

Chapter 20. Software Installer

Software Installer is an IBM product that supports software developers with a set of programs and functions for developing installation programs. The use of Software Installer will allow a standardized common way to install software products and will support manual and automatic software distribution and installation. It supports OS/2 and Windows operating systems. It provides a lot of functions, including the ability to CID-enable an installation program. It also includes a delete function which is a basic requirement to CID installations but not always followed. As it is already used by a lot of products, it might be useful to take a look at how products get installed if Software Installer was used to create the installation program. The products DB2/2 V2.11, Lotus Smart Suite, Faxworks, IBM Works and others already use those installation programs. If you want to have more information about Software Installer, check the manual *Software Installer*, SC34-4515 and the redbook *Examples using Software Installer*, GG24-2529. You can get the Software Installer code from the IBM Raleigh homepage at <http://installr.raleigh.ibm.com>, where the GA version of Software Installer is available for anyone to download. It is also available on the Developer Connection CDROM.

20.1 Files created by Software Installer

If the installation program of a product uses Software Installer, the files created to control the install are always the same. Right now, every program also brings the Software Installer files needed during the install with it. This might change with future versions of OS/2. The following files can be found:

- INSTALL.EXE

This is the installation procedure that will unpack the temporarily needed Software Installer files and starts the actual install program. It will also clean up the temporary directory after the install ends.

- INSTALL.IN_

This file holds the needed Software Installer files.

- *.PKG

This file holds a complete file list of the product that is to be installed. It may occur more than once if the developer decided to split its application in different parts, for example if there are various install modes of the product like in DB2/2 V2.11.

- *.ICF

This is the so called package file of a product, including a brief description.

If you find each of these files at least once, you have a product that is using an installation program created with Software Installer. If the developer has not explicitly permitted the unattended install the application will be installable in a CID environment.

20.2 Install Parameters

The following installation parameters are supported by the INSTALL.EXE:

- /A:

This parameter is for the action to be performed. Possible values are I for installation, U for update, D for delete and R for restore.

- /C:

This parameter defines the catalog file to be used. A fully qualified path is needed.

- /TU:

This parameter points to the drive where the CONFIG.SYS can be found that is to be updated.

- /L1:

This points to the error log of the install. A fully qualified path is needed.

- /L2:

This points to the history log of the install. A fully qualified path is needed.

- /P:

This is needed to define the name of the product to perform the action on.

- /R:

This parameter specifies the drive, path and file name of the response file.

- /G:

This parameter specifies the drive, path and file name of the general response files.

- /X

This is to indicate the installation program is to install the product in an unattended mode.

To find out if the parameters or values have changed, invoke INSTALL.EXE with an undefined parameter, like "?". You will then get an error message that includes a help button leading to the information for INSTALL.EXE.

20.3 Default Response File

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:

- OVERWRITE

This keyword specifies if existing files may be overwritten or not. The possible values are YES or NO.

- FILE

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.

- AUX1

This keyword specifies the boot drive.

- CFGUPDATE

This keyword specifies if CONFIG.SYS and AUTOEXEC.BAT are to be updated automatically during the install process. The possible values are YES, NO and AUTO.

- DELETEBACKUP

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.

Warning!

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.

20.4 Example for a Product that uses Software Installer

IBM Works, which is included in the Bonus Pack of OS/2 Warp V3, is using an installation procedure created with Software Installer. This is an example for a NetView DM/2 V2.1 profile for this product:

```
TargetDir=E:\

Section Catalog
Begin
    ObjectType=Software
    GlobalName=IBM.WORKS.INST.REF.1
    Description="Installation Procedure for IBM Works"
End

Section Install
Begin
    Program      = $(SourceDir)\INSTALL.EXE
    Pargs        = /A:I /S:$(SourceDir) /R:$(ResponseFile) /L1:$(LogFile1) /L2:$(LogFile2) /X
    ResponseFile = SA:\RSP\IBMWORKS\test.RSP
    SourceDir    = SA:\IMG\CONNECT\IBMWORKS
    LogFile1     = SB:\CONNECT\$(WorkStatName).wl1
    LogFile2     = SB:\CONNECT\$(WorkStatName).wl2
End
```

Figure 95. NetView DM/2 V2.1 Profile for IBM Works. Sample change file profile for IBM Works

If you want to install IBM Works in an LCU environment, you can use the following sequence for the product definition in the LCU command file:


```

i=i+1
x.ibmworks = i                                /* structure index */
x.i.name='OS/2 Warp Bonus Pak IBM Works '      /* product name */
x.i.statevar = ''                             /* state variable name */
x.i.instprog = 'x:\img\connect\ibmworks\install.exe ', /* install program name */
              ' /X ',                          /* Unattended mode flag */
              ' /A:I ',                        /* Action Flag: Install */
              ' /L1:\CONNECT\' || client || '.w11 ', /* errorlog file */
              ' /L2:\CONNECT\' || client || '.w12 ', /* history log */
              ' /R:'                           /* responsefile flag */
x.i.rspdir = 'x:\rsp\ibmworks'                 /* path to responsefile */
x.i.default = 'ibmworks.rsp'                   /* responsefile name */

```

Figure 96. LCU product definition sequence for IBM Works

This is the response file we used:

```

FILE = E:IBMWORKS
CFGUPDATE = AUTO
OVERWRITE = YES

```

Figure 97. IBMWORKS Response file.

20.5 Additional Information

Some more information on Software Installer that is not directly related to the CID install might also be useful.

Software Installer saves the information about all products using Software Installer for the install in two files, EPFIHCNF.CNF and EPFIS.INI. Both files are found in C:\OS2SYSTEM, assuming that OS/2 was installed on the C: drive. If these files get lost or corrupted, no update or delete of the installed products is possible. You will receive the error message that the related product is not installed. If you want to change the location of these files, you can use an environment variable:

```
SET EPFINSTDIR=D:CFG
```

and copy the files, if they already exist, to this subdirectory. You may also want to add these files to the critical system files that need to be backed up regularly.

20.6 Enabling a New product to Software Installer

This section shows the steps that are needed to CID-enable an application using Software Installer. It assumes that you either have a complete knowledge about what is done during the product's installation or that you want to use it for an application you wrote. This is not intended to replace the detailed information that may be found in: *Software Installer*, SC34-4515 and the redbook *Examples using Software Installer*, GG24-2529.

The following description is valid for Software Installer Version 1.3 and it assumes that Software Installer is already installed on the system.

- Open the **Software Installer** folder.

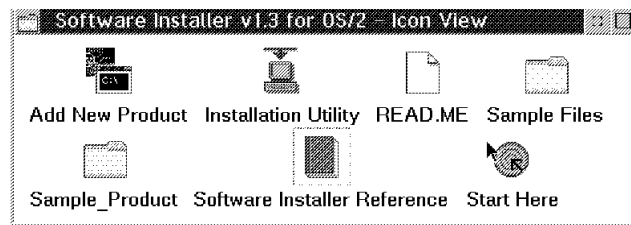


Figure 98. Software Installer folder.

- Select **Add new product** and give it a name when prompted.

This will create a new folder with the name of your product.

- Open the **product** folder you just created.

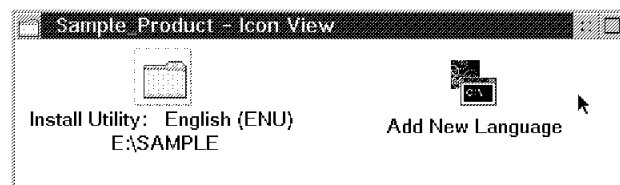


Figure 99. Software Installer sample product folder.

- Select **Add a New Language** and type ENU for English when prompted.
- Select **Install Utility** English(ENU).

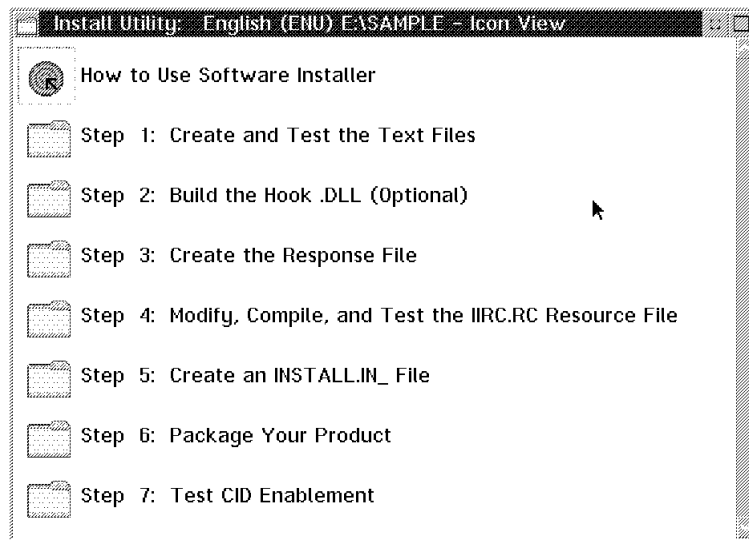


Figure 100. Software Installer install utility folder.

The Install Utility folder should have the following icons:

1. How to use Software Installer.
2. Step 1: Create and test text files.
3. Step 2: Build the Hook.DLL (optional).
4. Step 3: Create the Response file.
5. Step 4: Modify, Compile, and Test the IIRC.RC Resource File.
6. Step 5: Create an INSTALL.IN_ File.
7. Step 6: Package Your Product.
8. Step 7: Test CID Enablement.

20.6.1 Step by Step Description

- STEP 1

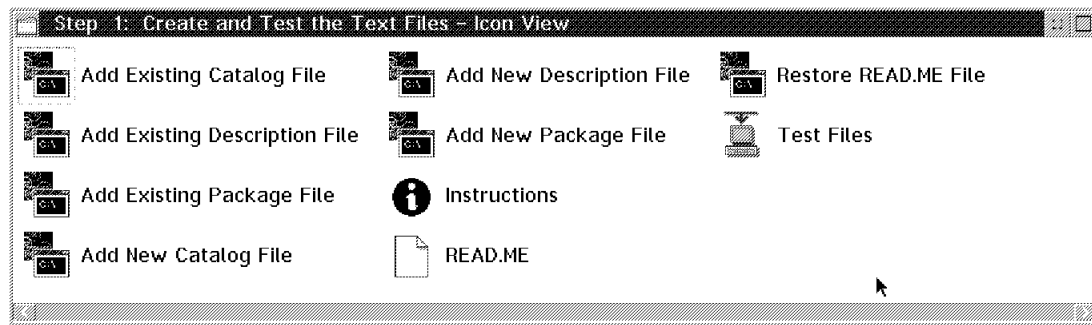


Figure 101. Software Installer Enabling new applications: Step1.

Use the Step 1 folder to create the catalog, package, description, and READ.ME Files for your product. You can use existing text files or use the templates and create new text files.

- Select **Add new Description File** and give it a name when prompted.
- Select **Add new Catalog File** and give it a name when prompted.
- Select **Add new Package File** and give it a name when prompted.
- Select **READ.ME** and modify it with your product information.
- Select **Test Files**.
- Select **Open** catalog from the file Pull-Down Menu.
- Select **Drive** from the open catalog Menu.
- Enter the drive that contains your catalog file.
- Enter the name of your catalog file in the file field.
- Select **Open**
- Select **Install** from the Action Menu.
- Check the Product number, Version, and Feature fields displayed in the Install window.
- Select **OK** to install the product.

When the Install - directories window is displayed, you have successfully tested the syntax of your catalog, package, and description.

• STEP 2

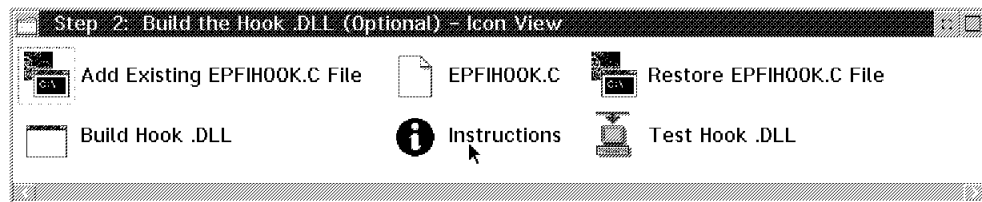


Figure 102. Software Installer Enabling new applications: Step2.

This step is optional. Use it if you want to change the processing of the install directories window.

Refer to the Software Installer Reference for information on using hooks to change installation directories.

- Select **EPFIHOOK.C**. You can use an existing EPFIHOOK.C file or modify the template. You can also refresh the template by selecting **Restore EPFIHOOK.C File**.
- Build the hook by selecting **Build Hook.DLL**.
- Test your modifications by selecting **Test Hook.DLL**.

• STEP 3

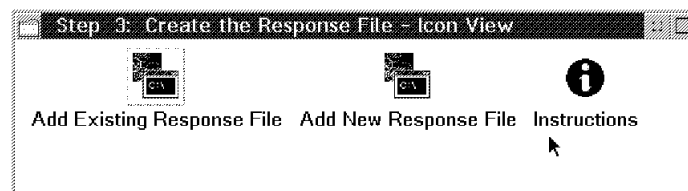


Figure 103. Software Installer Enabling new applications: Step3.

You can use response files to pass information to Software Installer executables. The response files must be on the workstation before the Software Installer executable (EPFIDLDS.EXE) is started.

Software Installer supports installations that have one specific response file and optionally have general response files that are included by the specific response file.

The response file must contain all the keywords or installation variables needed by your product's installation processing. Your documentation for unattended installation should explain how to set the values of the response file keywords. You can use an existing response file or use the template and create a new one.

- Select **Add New Response File** or **Add Existing Response File**.

- For a new response file, type a name and extension for your response file, for example **MYPROD.RSP**, and press **Enter**.

To use an existing response file, type the fully-qualified name, for example **C:\IBB\EPFISRSP.RSP**, and press enter.

- Press any key. Software Installer puts your file in the folder.
- Select the file. If you are creating a new file, follow the directions in the file to modify it for your product.
- Save the file and close it.

• STEP 4

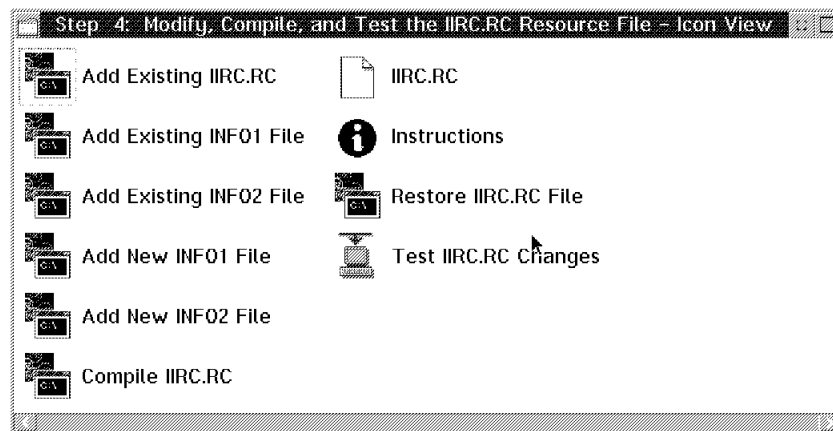


Figure 104. Software Installer Enabling new applications: Step4.

Use the **IIRC.RC** resource file to customize the appearance of your product installation.

- Select **IIRC.RC** file and modify it for your product. Then Save it and Close it.
- Select **Compile** to compile the IIRC.RC.
- Select **Test IIRC.RC Changes** to test the IIRC.RC

• STEP 5



Figure 105. Software Installer Enabling new applications: Step5.

The INSTALL.IN_ is a packed file containing your customized Software Installer program files plus other files specific to your application's installation.

- Select **Create INSTALL.IN_** to create an INSTALL.IN_ file containing the compressed software installer files and other files specific to your application.
- Press **Enter**. Software Installer creates the INSTALL.IN_ and places it in your working directory.

• **STEP 6**

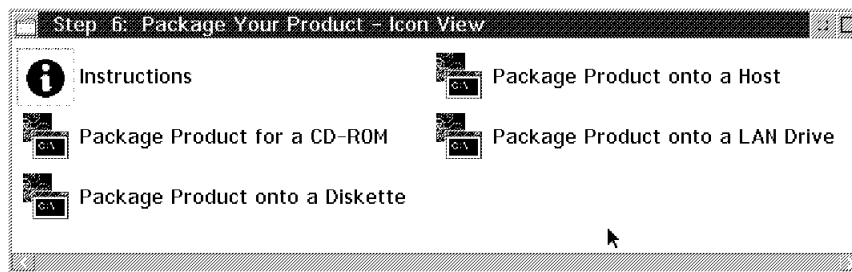


Figure 106. Software Installer Enabling new applications: Step6.

Step 6 contains four options.

1. **Diskette** Packaging Steps
2. **CD-ROM** Packaging Steps
3. **LAN** drive Packaging Steps
4. **Host** Packaging Steps

We will take #3 which is **LAN** packaging Steps. The rest of the Steps are described in the Start Here icon view.

- Select **Package Product onto a LAN drive**.

- Type the parameters using the following syntax: /<options>
<system> <input_package_file> <destination>
<source_directory and select **Open**.
- (Optional) Pack the output package file. You must use **EPFIPAK2.EXE** to pack the package file.
- Rename the output file using a .PKG (or .PK_, if packed) extension.

- **STEP 7**

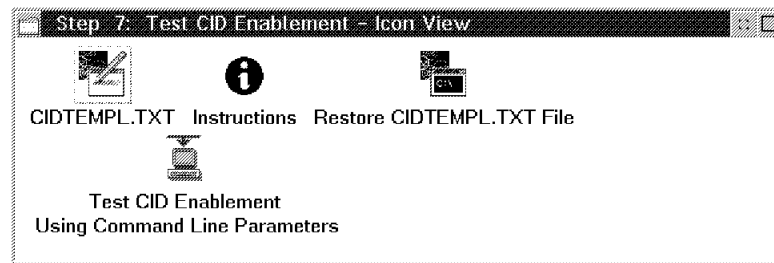


Figure 107. Software Installer Enabling new applications: Step7.

IBM has developed the CID method to standardize how products are installed and distributed. If your product must be CID certified, this step helps you ensure that you have met the requirements.

- Prepare your product documentation.
- Select **CIDTEMPL.TXT** and modify the file for your product.

CIDTEMP.TXT is a template that contains the CID information that you must provide for your users. Add this information to your product documentation. It must include:

- How to transfer diskette images to hard disk
- How to create a response file
- How to install your product using command line parameters
- What return codes your product uses
- Install your product using command line parameters by selecting **Test CID Enablement**.
- Specify your parameters and select **Open**.
- Install your product using **LCU**.
- Install your product using **NVDM/2**.

20.7 Example for non CID-Enabled Software

During our tests we installed MicroSoft Word V2.0 (a non CID-enabled product) in an unattended mode using NetView DM/2 V2.1 as the software distribution manager.

Microsoft Word 2.0 provides a special install mode called "Silent Setup" which can be used during an automated installation. This is not a CID-enabled installation mode, because it does not use input files like response files or log files. Therefore, it is necessary to check the updated WWORD20.INF file after doing the changes to it, in order to assure that the result of the install is the one that was expected. Extensive testing is necessary because the procedure and the setup program do not pass back any return codes.

We followed these steps:

- Install Microsoft Word as a server installation.
- Edit WWORD20.INF and uncomment everything under Silent Setup.
- Create a batch file.

This batch file assumes that LAN Requestor is already installed on the system, and that the product code for Microsoft Word 2.0 was put on LAN Server LS2. The server is accessed via LAN Logon and a NET USE command for the server's D: drive is issued. You might prefer creating an alias for Microsoft Word 2.0 and access it. The batch file has to be put on the NDM/2 CC server in the ShareDirA area, to a path that is referenced by the profile.

Sample of the batch

```
@echo off
logon userid /p:password /d:ls2dom /v=d
net use q: \\ls2\d$
q:
cd \winword
setup
```

- Create a change profile to run the batch.

Sample of Change File Profile

```
TargetDir=D:\winword
Section Catalog
Begin
    ObjectType = Software
    GlobalName = MSWord2.INST.REF.1
    Description = Install- Microsoft Word V2.0
End
Section Install
Begin
    Program = SA:\word.cmd
    Logfile1=SB:\LOG\Word\$(WorkStatName).11
    Logfile2=SB:\LOG\Word\$(WorkStatName).12
End
```

As seen in this example, it is useful to check the product information in detail even if it says nothing about CID enablement. There might be ways to make use of existing install programs before you start working with Software Installer to enable a product. These methods often lead to a simplified installation that cannot be customized per user.

Part 5. Appendixes

Appendix A. File Index Table

Some of the files listed below may be in a subdirectory on the sample CDROM. If you use

DIR drive:name.ext /S

all occurrences of the file 'name.ext' will be shown.

The table contains three columns. The first one is the name of the program or procedure. The next column specifies where the code can be found and the third is the reference to where the code is used or explained.

<i>Table 16 (Page 1 of 3). File Index Table.</i>		
Name	Code location	Usage
CASAGENT.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 104, 321
CASINSTL.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 100, 319
CASDELET.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 106
CRENVVAR.EXE	Sample CDROM - UTILITY	Page 557, 559
EXPORTS	TCPIPETC.on TCPIP server	Page 345
GETBOOT.CMD	NTS/2 Utilities diskette	Page 409
GETOSCID.CMD	NTS/2 Utilities diskette	Page 388
GETREXX.CMD	NTS/2 Utilities diskette	Page 410
HOSTS	TCPIPETC on TCPIP server	Page 346
IFSDEL.EXE	NTS/2 Utilities diskette	Page 98
LAPS.EXE	NTS/2 LAN Adapter and Protocol Support diskette	Page 98
LAPSDDEL.EXE	NTS/2 LAN Adapter and Protocol Support diskette, MPTS disk 1	Page 93
LAPSDISK.EXE	NTS/2 LAN Adapter and Protocol Support diskette, MPTS disk 1	Page 394
LAPSRSP.EXE	NTS/2 LAN Adapter and Protocol Support diskette, MPTS disk 1	Page 58, 305
MPTS.EXE	MPTS diskette 1	89, 305
NFSRFI.CMD	Sample CDROM - TCPIP	Page 349
NVDMBDSK.EXE	IBMNVDM2BIN of installed NetView DM/2	Page 369

<i>Table 16 (Page 2 of 3). File Index Table.</i>		
Name	Code location	Usage
NVDMCOPY.EXE	NetView DM/2 install disk 1	Page 403
NVDMPCSD.EXE	NetView DM/2 csd disk 2	Page 403
NVDMPMS.EXE	NetView DM/2 install disk 1	Page 403
NWDELETE.CMD	Sample CDROM - NETWARE	Page 130
NWINST.CMD	Sample CDROM - NETWARE	Page 128
REQDELE1.CMD	Sample CDROM - RIPL	Page 164
REQDL300.CMD	Sample CDROM - RIPL	Page 164
REQUPDAT.CMD	Sample CDROM - RIPL	Page 164
RSPINST.EXE	In REQUIRED bundle on OS/2 WARP Connect V3	Page 82, 387
SAMPLE.RSP	In REQUIRED bundle on OS/2 WARP Connect V3	Page 387, 591
SEDISK.EXE	In CID bundle on OS/2 WARP Connect V3	Page 385, 389, 591
SEIMAGE.EXE	In CID bundle on OS/2 WARP Connect V3	Page 385, 391
SEINST.EXE	In CID bundle on OS/2 WARP Connect V3	Page 79, 150, 157, 385
SEMAINT.EXE	In CID bundle on OS/2 WARP Connect V3	Page 86, 88, 183, 591
SERVICE.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 322
SERVICE.INI	NTS/2 Utilities diskette, MPTS disk 3	Page 323, 327, 619, 625
SHPIINST.DLL	In REQUIRED bundle on OS/2 WARP Connect V3	Page 385
SRVATTCH.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 97, 151, 626
SRVREXX.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 321
TCPSTART.CMD	Sample CDROM - TCPIP	Page 356
TCPDELET.CMD	Sample CDROM - TCPIP	Page 356
TCPINST.CMD	Sample CDROM - TCPIP	Page 123
TCPREP.CMD	Sample CDROM - TCPIP	Page 359
TCPSEED.CMD	Sample CDROM - TCPIP	Page 358
THINIFS.EXE	NTS/2 Utilities diskette, MPTS disk 3	Page 94, 158

<i>Table 16 (Page 3 of 3). File Index Table.</i>		
Name	Code location	Usage
THINLAPS.EXE	NTS/2 LAN Adapter and Protocol Support diskette	Page 91, 314, 317
THINR300.CMD	Sample CDROM - RIPL	Page 164
THINSRV.EXE	NTS/2 Utilities diskette	Page 311
THINTCP.CMD	Sample CDROM - TCPIP	Page 352

Appendix B. Versions Used in This Book

Below is a listing of the various software versions we used when doing the installations described in this book. Other versions can behave differently and have different requirements.

B.1.1.1 Software on the Code Servers

Only the software needed for a special type of code server was installed on it.

1. IBM Operating System/2 Warp Version 3
2. MPTS LAN Adapter and Protocol Support Version 5.00 (Syslevel WR08200)
LAN CID Utility Version 5.00 is delivered together with LAN Server V5.0.
3. IBM Communications Manager/2 Version 1.11, PC/3270 for OS/2 V4.1 and CM Server V4.0.
4. IBM DATABASE 2 for OS/2 Version 2.11
5. IBM Operating System/2 Local Area Network Server V5.0
6. IBM TCP/IP Version 3.0
7. Novell NetWare V4.1
8. IBM NetView Distribution Manager/2 Version 2.1 CSD Level XR00003

Using an MPTS LAN CID Utility code server using SRVIFS running on OS/2 WARP Connect V3 worked without problems for us. It is advisable to use the updated GETREXX.COM and GETBOOT.COM from the sample code CDROM otherwise not all of its necessary files are copied to the code server.

Since MPTS's LCU is being updated we recommend using it as soon as it becomes available.

B.1.1.2 Software that Was CID Installed

We tested the automated installation for the following products in the CID environment.

1. IBM Operating System/2 Version 2.11.
2. IBM Communications Manager/2 Version 1.11
3. PC/3270 for OS/2 V4.1
4. CM Server V4.0

5. IBM DATABASE 2 for OS/2 Version 2.11
6. IBM Operating System/2 Local Area Network Server V5.0
7. IBM TCP/IP Version 3.0
8. Novell Netware Workstation for OS/2 V2.11
9. IBM NetView Distribution Manager/2 Version 2.1 CSD Level XR0003

The referenced Service Paks are the U.S. versions. If you are using a national language version you will need the corresponding national language version of the Service Pak.

Appendix C. OS/2 Response File Keywords

The following is a list of all KEYWORDS in the response file that are supported by the OS/2 installation process:

Table 17 (Page 1 of 4). OS/2 Response File Keywords Table. The following table contains the OS/2 response file keywords and indications as to what versions they are valid for.				
Keyword	OS/2 V2.11	OS/2 Warp V3	OS/2 Warp with WinOS2 V3	OS/2 WARP Connect V3
AdditionalPrinters		√	√ 1	√
AlternateAdapter	√	√	√	√
APM	√	√	√	√
BaseFileSystem	√	√		√
CDROM	√	√ 1	√ 1	√
ConfigSysLine	√	√	√	√
Copy	√	√	√	√
CountryCode	√ 1	√ 1	√	√
CountryKeyboard	√ 1	√ 1	√	√
DDIDDP	√	√	√	√
DDIDest	√	√	√	√
DDISrc	√	√	√	√
DefaultPrinter	√ 1	√ 1	√ 1	√
DiagnosticAids	√	√	√	√
DisplayAdapter	√	√	√	√

Table 17 (Page 2 of 4). OS/2 Response File Keywords Table. The following table contains the OS/2 response file keywords and indications as to what versions they are valid for.

Keyword	OS/2 V2.11	OS/2 Warp V3	OS/2 Warp with WinOS2 V3	OS/2 WARP Connect V3
Documentation	√	√	√	√
DOSSupport	√	√	√	√
DPMI	√	√	√	√
EarlyUserExit	√	√	√	√
ExistingWindowsPath	√	√	√	√
ExitOnError	√	√	√	√
ExtendedInstall	√	√	√	√
Fonts	√	√	√	√
FormatFAT		√	√	√
FormatHPFS		√	√	√
FormatPartition	√	√		√
ID	√		√	√
Include	√	√	√	√
IncludeAtEnd	√	√	√	√
IncludeInLine	√	√	√	√
MigrateApplications	√	√	√	√
MigrateConfigFiles	√	√	√	√
MoreBitmaps	√	√	√	√

Table 17 (Page 3 of 4). OS/2 Response File Keywords Table. The following table contains the OS/2 response file keywords and indications as to what versions they are valid for.

Keyword	OS/2 V2.11	OS/2 Warp V3	OS/2 Warp with WinOS2 V3	OS/2 WARP Connect V3
Mouse	√	√	√	√
MousePort	√	√	√	√
MultimediaSupport		√	√	√
OptionalFileSystem	√	√	√	√
OptionalSystemUtilities	√	√	√	√
OS2IniData	√	√	√	√
PCMCIA	√	√ 1	√ 1	√
PCMCIAOptions		√	√	√
PrimaryCodePage	√	√	√	√
PrinterPort	√	√	√	√
ProcessEnvironment	√	√	√	√
ProgressIndication	√	√	√	√
RebootRequired	√	√	√	√
REXX	√			
SCSI	√	√	√	√
SeedConfigSysLine	√	√	√	√
SerialDeviceSupport	√	√	√	√
ShareDesktopConfigFiles	√		√	√

Table 17 (Page 4 of 4). OS/2 Response File Keywords Table. The following table contains the OS/2 response file keywords and indications as to what versions they are valid for.

Keyword	OS/2 V2.11	OS/2 Warp V3	OS/2 Warp with WinOS2 V3	OS/2 WARP Connect V3
SourcePath	√	√	√	√
TargetDrive	√	√	√	√
ToolsAndGames	√	√ 1	√	√
UserExit	√	√	√	√
Version	√		√	√
WindowsInstallSourcePath		√		√
WindowsSupport		√		√
WIN-OS/2Desktop	√		√	√
WIN-OS/2Support	√		√	√
WIN-OS/2TargetDrive	√		√	√
WindowedWIN-OS/2	√		√	√
Note: 1 Valid keyword values have changed between versions.				

C.1 Keyword Description

The following is a short description of all the keywords and valid entries that can be used in a response file.

- **AdditionalPrinters** (Supported in OS/2 Warp V3, OS/2 Warp with WinOS2 V3, OS/2 WARP Connect V3)

Allows additional printers other than the default printer, see on page 446, to be installed.

KEYVALUE= 0=None
or
m1:n1,m2:n2, ...

Where m1, m2, etc. are port numbers as defined under "PrinterPort" below, and n1, n2, etc. are indexes into PRDESC.LST as described in C.3, "Printer Description Table for AdditionalPrinters and DefaultPrinter Keywords" on page 462.

Example: AdditionalPrinters=2:150

This will put an IBM 2380 PPS II on LPT2:. The 150 specifies the PPS II (if it is line 150 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, Not supported in OS/2 V2.0)**

Specifies whether or not to install APM.

0 = Don't install
1 = Autodetect (DEFAULT)
2 = Install

- **BaseFileSystem (Not supported in OS/2 Warp with WinOS2 V3)**

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 version dependent.

For OS/2 V2.11, these are:

- 0 = None
- 1 = Autodetect
- 2 = CDTechnology T3301, T3401
- 3 = Chinon431, 435
- 4 = Chinon535
- 5 = CreativeLabs OmniCD
- 6 = Hitachi1650,1750S,3650
- 7 = Hitachi1950S,3750,6750
- 8 = IBMCD-ROM I
- 9 = IBMCD-ROM II, Enhanced CD-ROM II
- 10 = IBMISA CD-ROM
- 11 = MitsumiCRMC-LU002S
- 12 = MitsumiCRMC-LU005S
- 13 = MitsumiCRMC-FX001
- 14 = MitsumiCRMC-FX001D
- 15 = NECIntersect 25,36,37,72,73,74,82,83,84
- 16 = NECMultiSpin 3Xi,3Xe,3Xp,38,74-1,84-1
- 17 = Panasonic501,LK-MC501S
- 18 = Panasonic521,522,523
- 19 = Panasonic562,563
- 20 = PhilipsLMS CM-215
- 21 = PioneerDRM-600
- 22 = PioneerDRM-604X
- 23 = SonyCDU-31A,33A,7305
- 24 = Sony541,561,6211,7211,7811
- 25 = Sony6111
- 26 = Texel3021,5021
- 27 = Texel3024,3028,5024,5028
- 28 = Toshiba3201
- 29 = Toshiba3301,3401,4101
- 30 = OTHER

For OS/2 Warp V3, these are:

- 0 = None
- 1 = Autodetect
- 2 = CDTechnology T3301, T3401
- 3 = Chinon431, 435
- 4 = Chinon535
- 5 = CompaqDual Speed
- 6 = CreativeLabs OmniCD
- 7 = Hitachi1650S,1750S,3650
- 8 = Hitachi1950S,3750,6750
- 9 = IBMCD-ROM I
- 10 = IBMCD-ROM I rev 242
- 11 = IBMCD-ROM II, Enhanced CD-ROM II
- 12 = IBMISA,Panasonic 562,563
- 13 = MitsumiCRMC-LU002S,Tandy CDR-1000
- 14 = MitsumiCRMC-LU005S
- 15 = MitsumiCRMC-FX001
- 16 = MitsumiCRMC-FX001D
- 17 = MitsumiCRMC-FX001DE
- 18 = NECIntersect 25,36,37,72,73,74,82,83,84
- 19 = NECMultiSpin 4Xe,4xi,3Xi,3Xe,3Xp,38,74-1,84-1
- 20 = NEC2vi,260
- 21 = Panasonic501,LK-MC501S
- 22 = Panasonic521,522,523
- 23 = PhilipsLMS CM-205,CM-225
- 24 = PhilipsLMS CM-205MS,206,225MS,226
- 25 = PhilipsLMS CM-215
- 26 = PhilipsLMS CM-207
- 27 = PioneerDRM-600
- 28 = PioneerDRM-604X
- 29 = PlectorDM-3028,DM-5028,4PLEX
- 30 = SonyCDU-31A,33A,7305,7405
- 31 = SonyCDU-531,535,6150,6201,6205,6251,7201,7205
- 32 = SonyCDU-55D,55E
- 33 = Sony541,561,6211,7211,7811
- 34 = Sony6111
- 35 = Texel3021,5021
- 36 = Texel3024,3028,5024,5028
- 37 = Toshiba3201
- 38 = Toshiba3301,3401,4101
- 39 = WearnesCDD-120
- 40 = Non-listedIDE CD-ROM
- 41 = OTHER

For OS/2 Warp with WinOS2 V3, these are:

- 0 = None
- 1 = Autodetect
- 2 = AztechCDA-268-03I-SE
- 3 = CDTechnology T3301, T3401
- 4 = Chinon525I
- 5 = Chinon431, 435
- 6 = Chinon535
- 7 = CompaqTray Load
- 8 = CompaqDual Speed
- 9 = CreativeLabs OmniCD
- 10 = GoldstarGCD-R520B
- 11 = Hitachi1650S,1750S,3650
- 12 = Hitachi1950S,3750,6750
- 13 = IBMCD-ROM I
- 14 = IBMCD-ROM I rev 242
- 15 = IBMCD-ROM II, Enhanced CD-ROM II
- 16 = IBMISA,Panasonic 562,563
- 17 = LionOptics XC-200AI,200EI
- 18 = MitsumiCRMC-LU002S,Tandy CDR-1000
- 19 = MitsumiCRMC-LU005S
- 20 = MitsumiCRMC-FX001
- 21 = MitsumiCRMC-FX001D
- 22 = MitsumiCRMC-FX001DE,FX300,FX400
- 23 = NECIntersect 25,36,37,72,73,74,82,83,84
- 24 = NECMultiSpin 4Xe,4xi,3Xi,3Xe,3Xp,38,74-1,84-1
- 25 = NEC2vi,260
- 26 = OpticsStorage 8001 IDE
- 27 = PanasonicCF-41
- 28 = Panasonic501,LK-MC501S
- 29 = Panasonic521,522,523
- 30 = Panasonic571
- 31 = PhilipsLMS CM-205,CM-225
- 32 = PhilipsLMS CM-205MS,206,225MS,226
- 33 = PhilipsLMS CM-215
- 34 = PhilipsLMS CM-207
- 35 = PioneerDRM-600
- 36 = PioneerDRM-604X
- 37 = PlectorDM-3028,DM-5028,4PLEX
- 38 = SanyoCRD-450P
- 39 = SonyCDU-31A,33A,7305,7405
- 40 = SonyCDU-531,535,6150,6201,6205,6251,7201,7205
- 41 = SonyCDU-55D,55E,76E
- 42 = Sony541,561,6211,7211,7811
- 43 = Sony6111
- 44 = Texel3021,5021
- 45 = Texel3024,3028,5024,5028

46 = ThinkPad 755CD, Teac CD-40E
47 = Toshiba3201
48 = Toshiba3301,3401,4101,3501,5201
49 = Toshiba5302B
50 = WearnesCDD-120
51 = Non-listedIDE CD-ROM
52 = OTHER

For OS/2 WARP Connect V3, these are:

0 = None
1 = Autodetect
2 = AztechCDA-268-03I-SE
3 = CDTechnology T3301, T3401
4 = Chinon525I
5 = Chinon431, 435
6 = Chinon535
7 = CompaqTray Load
8 = CompaqDual Speed
9 = CreativeLabs OmniCD
10 = GoldstarGCD-R520B
11 = Hitachi1650S,1750S,3650
12 = Hitachi1950S,3750,6750
13 = IBMCD-ROM I
14 = IBMCD-ROM I rev 242
15 = IBMCD-ROM II, Enhanced CD-ROM II
16 = IBMISA,Panasonic 562,563
17 = LionOptics XC-200AI,200EI
18 = MitsumiCRMC-LU002S,Tandy CDR-1000
19 = MitsumiCRMC-LU005S
20 = MitsumiCRMC-FX001
21 = MitsumiCRMC-FX001D
22 = MitsumiCRMC-FX001DE,FX300,FX400
23 = NECIntersect 25,36,37,72,73,74,82,83,84
24 = NECMultiSpin 4Xe,4xi,3Xi,3Xe,3Xp,38,74-1,84-1
25 = NEC2vi,260
26 = OpticsStorage 8001 IDE
27 = PanasonicCF-41
28 = Panasonic501,LK-MC501S
29 = Panasonic521,522,523
30 = Panasonic571
31 = PhilipsLMS CM-205,CM-225
32 = PhilipsLMS CM-205MS,206,225MS,226
33 = PhilipsLMS CM-215
34 = PhilipsLMS CM-207
35 = PioneerDRM-600
36 = PioneerDRM-604X

37 = PlextorDM-3028,DM-5028,4PLEX
 38 = SanyoCRD-450P
 39 = SonyCDU-31A,33A,7305,7405
 40 = SonyCDU-531,535,6150,6201,6205,6251,7201,7205
 41 = SonyCDU-55D,55E,76E
 42 = Sony541,561,6211,7211,7811
 43 = Sony6111
 44 = Texel3021,5021
 45 = Texel3024,3028,5024,5028
 46 = ThinkPad 755CD, Teac CD-40E
 47 = Toshiba3201
 48 = Toshiba3301,3401,4101,3501,5201
 49 = Toshiba5302B
 50 = WearnesCDD-120
 51 = Non-listedIDE CD-ROM
 52 = OTHER

- **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.

KEYVALUE=a valid CONFIG.SYS statement

- **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:OS2INSTALL. 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.

Note: 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.

KEYVALUE=sourcefile destination

source file = valid filename
 destination = valid directory name

Example: Copy = c:\text.dat z:\

- **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. Note that for some countries support has been added in OS/2 Warp V3 and there might not be support for some countries in WIN-OS2 (then they are defined there as "other country").

Country code	Country	Codepage
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. 2-5 character keyboard code.

IMPORTANT

The keyboard codes for FR, IT and UK keyboards are different between OS/2 V2.11, OS/2 Warp V3 and OS/2 WARP Connect V3. Use appropriate code as shown in the following list.

AR = Arabic
BE = Belgium
BR = Brazil
CF = Canada,
CS243 = Czechos
CS245 = Czechos
DK = Denmark
SU = Finland
FR = France
FR189 = France
FR120 = France,
GR = Germany
HE = Hebrew
HU = Hungary
IS = Iceland
IT = Italy
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
UK = United Kingdom
UK166 = United Kingdom
UK168 = United Kingdom
US = United States
YU = Yugoslavia

- **DDIDDP (Not supported in OS/2 V2.0)**

Use OS/2 Device Driver Installation 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.

KEYVALUE=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 Appendix I, "Hardware and Software Dependencies" on page 571. 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 (hit Esc key).
- Create a new subdirectory in the IMGOS2V211 or the IMGOS2V3 directory on the code server.
- Copy from the Backup reference diskette BIOS.SYS, .BIO and .DDP files into that directory.
- Reference this directory by the DDISrc keyword.

- **DDIDest (Not supported in OS/2 V2.0)**

The OS/2 Device Driver Installation Destination. This determines the target directory for the device driver. (See also **DDIDDP** and **DDISrc**.)

KEYVALUE=Directory where to copy the device driver files.

- **DDISrc (Not supported in OS/2 V2.0)**

The OS/2 Device Driver Installation Source. This determines the source directory for the .DDP files. (See also **DDIDDP** and **DDISrc**.)

KEYVALUE=Directory where the .DDP files are

Using DDInstall for ABIOS

Using DDInstall, the response file has to have the following entries:

```
DDISrc   = Z:IMGCONNECTABIOSPC700
DDIDest  = C:\
DDIDDP   = ABIOS.DDP
```

Please keep in mind that you need a fully qualified path to the DDISrc, even when using NDM/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 ABIOS files on a system without a reference partition. Therefore, double-check the paths entered here, and the file name for the DDP file.

- **DefaultPrinter**

Specifies which default printer to install.

KEYVALUE=Keyvalue in the corresponding printer table

0 = None

See C.3, "Printer Description Table for AdditionalPrinters and DefaultPrinter Keywords" on page 462 for an explanation of how to find the keyvalue.

- **DiagnosticAids**

Specifies whether or not to install certain Reliability, Availability and Serviceability utilities.

0 = Don't install

1 = Install (DEFAULT)

- **DisplayAdapter**

Specifies which adapter should override the primary adapter detected by the install process.

- 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

- **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 (Not supported in OS/2 V2.0)**

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.

KEYVALUE=user exit program name (DEFAULT=none)

For an example on the use of EarlyUserExit refer to C.9, "The User Exit Keywords of the Response File" on page 480.

- **ExistingWindowsPath**

Specifies the path to an existing Windows system.

For OS/2 2.x this option is valid only when option 1 is selected for the WIN-OS/2 Desktop keyvalue.

For OS/2 Warp V3 if WindowsSupport is selected and this value is NULL (not set), install will search all partitions for existing Windows system and the first valid Windows 3.1 system will be selected. If a Windows 3.1 system is not found, an error will be generated and response file install will be aborted.

KEYVALUE=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 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.

KEYVALUE=full pathname 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 (Supported in OS/2 Warp V3, OS/2 Warp with WinOS2 V3, OS/2 WARP Connect V3)**

Specifies which drives to format with FAT file system. This keyword cannot be used in conjunction with the BaseFileSystem or FormatPartition keywords.

KEYVALUE=Drives to format as FAT

example: FormatFAT=C:,D:,E:

- **FormatHPFS (Supported in OS/2 Warp V3, OS/2 Warp with WinOS2 V3, OS/2 WARP Connect V3)**

Specifies which drives to format with HPFS file system. This keyword cannot be used in conjunction with the BaseFileSystem or FormatPartition keywords.

KEYVALUE=Drives to format as HPFS

example: FormatHPFS=F:,G:

- **FormatPartition**

Specifies whether or not to format the install partition, using 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

- **ID (Only supported in OS/2 V2.11)**

Specifies some identification string which may be used by install or UserExit to identify the response file(s) used for this installation. This keyword is user defined.

This keyword is not used with OS/2 Warp V3.

KEYVALUE=ASCII string

- **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.

KEYVALUE=valid filename

For an example of the use of Include within a response file see C.8.1, "Single Include and IncludeAtEnd File Example" on page 476.

- **IncludeAtEnd (Not supported in OS/2 V2.0)**

Specifies another response file to process along with the current one. There may be multiple occurrences of this keyword. The "Included" response file is appended to the end of all response files that have been processed before this one. The result of IncludeAtEnd is the same as if Include is used.

The fully qualified path is required when specifying an include file.

KEYVALUE=valid filename

- **IncludeInLine (Not supported in OS/2 V2.0)**

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.

KEYVALUE=valid filename

For an example of the use of IncludeInLine within a response file see C.8.2, "Single IncludeInLine File Example" on page 477.

- **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.

KEYVALUE=Drives to search, database 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)

- **MoreBitmaps**

Specifies whether or not to install more bitmaps for the Wallpaper utility.

0 = Don't install More Bitmaps

1 = Install More Bitmaps (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 (Supported in OS/2 Warp V3, OS/2 Warp with WinOS2 V3, OS/2 WARP Connect V3)**

Specifies whether or not to install multimedia files during the installation. No configuration is supported at this point. For further information see 4.1.1.4, “Multimedia Support” on page 86.

0 = do NOT install multimedia support
1 = install multimedia support (DEFAULT)

Example: MultimediaSupport=1

- **OptionalFileSystem**

Specifies whether or not to install optional file system(s). This option is provided so that if you initially decided to install OS/2 using the FAT file system, OS/2 will not copy any of the HPFS files to your disk. The user might require the use of these HPFS files, and can therefore have the ability to install both file systems.

For example the user initially only has one drive C: using FAT. At a later stage the user decides to add a extra partition D: using HPFS. Using this option, all the necessary DLL files are copied to disk.

0 = Do Not Install Optional File System(s)
1 = Install Optional File System (DEFAULT)

- **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

- **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 *PM*

Programming Reference in the chapter "Profile Functions". Valid keywords are found in the Appendix "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.

KEYVALUE=/AppName/KeyName/KeyValue/

NOTE: Since each of these names can contain imbedded blanks and whitespace, 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

Which would define the default queue name.

- **PCMCIA (Not supported in OS/2 V2.0)**

Specifies whether or not to install PCMCIA. For OS/2 V2.11 the following are valid values:

0 = Don't install
1 = Install (DEFAULT)

For OS/2 Warp V3 the following are valid values:

0 = Don't install (DEFAULT)
1 = Ambra
2 = ASTBravo
3 = ASTPowerExec
4 = CompaqConcerto
5 = Compuadd425TX
6 = IBMThinkPad 350
7 = IBMThinkPad 360
8 = IBMThinkPad 500
9 = IBMThinkPad 510
10 = IBMThinkPad 720
11 = IBMThinkPad 750
12 = IBMThinkPad 755
13 = IBMPS/2 E
14 = Matsushita
15 = NCRSafari
16 = NECVersa
17 = Panasonic

18 = ToshibaT3600
19 = ToshibaT4500
20 = ToshibaT4600
21 = ToshibaT4700
22 = ToshibaT4800
23 = Zeos
24 = ZenithZ-lite 425L

For OS/2 WARP Connect V3 the following are valid values:

0 = Don't install (DEFAULT)
1 = Ambra486 SN425C
2 = ASTAscentia 800N
3 = ASTBravo
4 = ASTPowerExec
5 = AustinDSTN
6 = CompaqConcerto
7 = Compuadd425TX
8 = DELLLatitude
9 = DELLLatitude XP
10 = IBMThinkPad 230
11 = IBMThinkPad 350
12 = IBMThinkPad 360
13 = IBMThinkPad 500
14 = IBMThinkPad 510
15 = IBMThinkPad 701
16 = IBMThinkPad 720
17 = IBMThinkPad 750
18 = IBMThinkPad 755 C/CS
19 = IBMThinkPad 755 CE/CSE
20 = IBMThinkPad 755 CD
21 = IBMPS/2 E
22 = Matsushita
23 = NCRSafari
24 = NECVersa
25 = Panasonic
26 = ToshibaT3600
27 = ToshibaT4500
28 = ToshibaT4600
29 = ToshibaT4700
30 = ToshibaT4800
31 = Zeos
32 = ZenithZ-lite 425L

- **PCMCIAOptions (Supported in OS/2 Warp V3, OS/2 Warp with WinOS2 V3, OS/2 WARP Connect V3)**

0=Don't install (DEFAULT)
1=Install all
2=Modem/FAX services
3=Hard disk services
4=FLASH services

- **PrimaryCodePage**

Specifies whether "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 443. 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 primitive programs, batch files, etc. (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.

- **REXX (Keyword not supported in OS/2 Warp V3 or OS/2 Warp with WinOS2 V3, OS/2 WARP Connect V3)**

Specifies whether or not to install REXX. For OS/2 Warp V3 and OS/2 Warp with WinOS2 V3 there is no REXX keyword, since REXX will always be installed.

0 = Don't Install REXX
1 = Install REXX (DEFAULT)

- **SCSI (Not supported in OS/2 V2.0)**

Specifies which, if any, SCSI adapter device you wish to install support for.

Valid values for OS/2 V2.11 are:

0 = None
1 = Autodetect
2 = Adaptec1510, 1520, 1522
3 = Adaptec1540, 1542
4 = Adaptec1640
5 = Adaptec1740, 1742, 1744
6 = BusLogicBusMaster SCSI Adapters
7 = DPTPM2011, PM2012
8 = FutureDomain 845,850,850IBM,860,875,885
9 = FutureDomain 1650,1660,1670,1680,MCS700
10 = FutureDomain 7000EX
11 = IBMPS/2 SCSI Adapter
12 = IBM16-Bit AT Fast SCSI Adapter

Valid values for OS/2 Warp V3 are:

0 = None
 1 = Autodetect
 2 = Adaptec1510,1520,1522
 3 = Adaptec1540,1542
 4 = Adaptec1640
 5 = Adaptec1740,1742,1744
 6 = Adaptec2840VL,2842VL,2740,2742,AIC7770
 7 = Adaptec2940,2940W,AIC7870
 8 = BusLogicBusMaster SCSI Adapters
 9 = DPTPM2011,PM2012
 10 = FutureDomain 845,850,850IBM,860,875,885,TMC 9C50/C950
 11 = FutureDomain 16xx,1790,1795,MCS600/700,TMC 1800/18C30/18C50/3260/36C70
 12 = FutureDomain 7000EX
 13 = IBMPS/2 SCSI Adapter
 14 = IBM16-Bit AT Fast SCSI Adapter
 15 = ProAudioSpectrum 16 with Trantor SCSI

Valid values for OS/2 WARP Connect V3 are:

0 = None
 1 = Autodetect
 2 = Adaptec1510,1520,1522
 3 = Adaptec1540,1542
 4 = Adaptec1640
 5 = Adaptec1740,1742,1744
 6 = Adaptec2840VL,2842VL,2740,2742,AIC7770
 7 = Adaptec2940,2940W,AIC7870
 8 = BusLogicBusMaster SCSI Adapters
 9 = DPTPM2011,PM2012
 10 = FutureDomain 845,850,850IBM,860,875,885,TMC 9C50/C
 11 = FutureDomain 16xx,1790,1795,MCS600/700,TMC 1800/18
 12 = FutureDomain 7000EX
 13 = IBMPS/2 SCSI Adapter
 14 = IBM16-Bit AT Fast SCSI Adapter
 15 = ProAudioSpectrum 16 with Trantor SCSI

- **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.

KEYVALUE=a valid CONFIG.SYS statement

- **SerialDeviceSupport**

Specifies whether or not to install the device driver.

0 = Don't install
1 = Install (DEFAULT)

- **ShareDesktopConfigFiles (Not supported in OS/2 Warp V3)**

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/2 Desktop keyvalue.

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, as 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.

KEYVALUE=drive and optional path (D:IMGOS2V211...)
DEFAULT=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.

KEYVALUE=C:

- **ToolsAndGames**

Specifies which tools or games can be installed. For OS/2 V2.11 the following values are valid:

0 = Install none
1 = Install all (DEFAULT)
2 = Enhanced Editor
3 = Search and Scan Tool
4 = Terminal Emulator
5 = Chart Maker
6 = Personal Productivity
7 = Solitaire - Klondike
8 = Reversi
9 = Scramble

10 = Cat and Mouse
11 = Pulse
12 = Jigsaw
13 = Chess

For OS/2 Warp V3, OS/2 WARP Connect V3 the following values are valid:

0 = Install none
1 = Install all (DEFAULT)
2 = Enhanced Editor
3 = Search and Scan Tool
4 = Solitaire - Klondike
5 = Pulse
6 = Chess
7 = Mahjongg Solitaire

example: ToolsAndGames=2,5,6

- **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.

KEYVALUE=user exit program name (DEFAULT=none)

- **Version (Only supported Until OS/2 V2.11)**

Specifies specific version of the operating system for which this file is intended. This keyword is user defined.

KEYVALUE=User defined version string

- **WindowsInstallSourcePath (Supported in OS/2 Warp V3 only)**

Specifies the path to Windows diskettes in a CID directory tree. When installing OS/2 Warp V3 on top of an **existing** DOS and Windows installation the OS/2 Warp V3 installation program requires some files from the Windows installation diskettes if the WindowsSupport keyword is set to 1.

The WindowsInstallationSourcePath ensures that these files can be found without manual intervention.

KEYVALUE= A string that specifies the path relative to the source path where the windows diskettes reside of the CID tree.

example: WindowsInstallSourcePath=\WIN31\DISKETTES

In the above example if the SourcePath keyword is set to Z:IMGOS2V30 this would cause the install program to look for the Windows diskette 1 in the Z:IMGOS2V30WIN31DISKETTESDISK_W1 directory.

- **WindowsSupport (OS/2 Warp V3 only)**

Specifies whether or not to support Windows 3.1. OS/2 Warp V3 can be installed on top of Windows 3.1 and 3.11 or Windows for Workgroups 3.1 and 3.11.

If the user wishes to change Windows version this should be done prior to installing OS/2 Warp V3. Otherwise OS/2 Warp V3 has to be reinstalled after the change of Windows version.

0 = Don't support Windows 3.1
1 = Support Windows 3.1

Example: WindowsSupport=1

- **WIN-OS/2Desktop (Supported in OS/2 V2.11, OS/2 Warp with WinOS2 V3)**

Specifies what the WIN-OS/2 desktop should look like. Option 1 should be selected only if Windows currently exists (See **ExistingWindowsPath** and **ShareDesktopConfigFiles** also). 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 (Supported in OS/2 V2.11, OS/2 Warp with WinOS2 V3)**

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 (Supported in OS/2 V2.11, OS/2 Warp with WinOS2 V3)**

Specifies on which valid partition drive to install WIN-OS/2.

KEYVALUE=any valid FORMATTED partition.

C: (DEFAULT)

D:

.

.

Z:

Example: WIN-OS/2TargetDrive=D:

would install WIN-OS/2 to partition D: located in
\OS2\MDOS\WINOS2

- **WindowedWIN-OS/2 (Supported in OS/2 V2.11, OS/2 Warp with WinOS2 V3)**

Specifies whether Windows applications should run in windowed sessions on the Presentation Manager desktop or in full screen sessions. Systems with (VGA) resolution will always receive WIN-OS/2 sessions that run in a window as the default.

0=Windowed WIN-OS/2 sessions (Requires medium resolution (VGA) video)
1=Full Screen WIN-OS/2 sessions (Run with highest resolution video possible)

C.2 How to Edit the Response File

- The response file is a flat ASCII file so it can easily be edited and manipulated.
- Comments may appear anywhere in the file. The comment character is the asterisk "*". Any line beginning with this character, will be ignored.

Note

Comments are full line comments and cannot be attached to the end of a line starting with a keyword.

- Blank lines are ignored.
- All OS/2 System Installation process response statements must have the following format:

KEYWORD=parm,parm,parm

If the keyword has the option to choose more than one parameter, the user can do so as long as they are separated by a comma, and do NOT include the default parameter.

- Statements do not need to start in the first column.
- The keywords may appear in any order. If a duplicate keyword exists the last one found will be used.
- keywords and "parm" values are not case sensitive.
- Blanks and white spaces on any lines are ignored in the keyword portion of the line. This is the portion preceding the delimiter "=".
- Blanks and white spaces are preserved in the "parm" portion of a response line. This is the portion following the delimiter "=".
- "parm" is an ASCII-numeric value (wherever possible) or a file specification to avoid typing problems and translation problems.
- The entire response file is processed before the rest of the installation occurs.

C.3 Printer Description Table for AdditionalPrinters and DefaultPrinter Keywords

A list of printers (PRDESC.LST) resides on the first printer device driver diskette. On an installed system it can be found in the OS2INSTALL directory. Please remember that this list is version dependent since more printer drivers are added with each upgrade. The **DefaultPrinter** keyvalue is used as an index in the list.

DefaultPrinter example

DefaultPrinter = 5 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)
Agfa Matrix ChromaScript v51_8: Agfa Matrix ChromaScript v51_8 (PSCRIPT.DRV)
Agfa-Compugraphic 9400PS v49_3: Agfa-Compugraphic 9400PS v49_3 (PSCRIPT.DRV)
Agfa/Compugraphic 400PS: Agfa/Compugraphic 400PS (PSCRIPT.DRV)
5 -----> Apple LaserWriter: Apple LaserWriter (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)
Apple LaserWriter Plus v42_2: Apple LaserWriter Plus v42_2 (PSCRIPT.DRV)
COMPAQ PAGEMARQ 15: COMPAQ PAGEMARQ 15 (PSCRIPT.DRV)
COMPAQ PAGEMARQ 20: COMPAQ PAGEMARQ 20 (PSCRIPT.DRV)
Citizen PN48: Citizen PN48 (EPSON.DRV)
Colormate PS v51_9: Colormate PS v51_9 (PSCRIPT.DRV)
Dataproducts LZR 1260 v47_0: Dataproducts LZR 1260 v47_0 (PSCRIPT.DRV)
Dataproducts LZR-2665: Dataproducts LZR-2665 (PSCRIPT.DRV)
Digital LN03R ScriptPrinter: Digital LN03R ScriptPrinter (PSCRIPT.DRV)
Digital LPS PrintServer 40: Digital LPS PrintServer 40 (PSCRIPT.DRV)
Epson 24 pins - 136 columns: 24-pin 136 Col (EPSON.DRV)
Epson 24 pins - 80 columns: 24-pin 80 Col (EPSON.DRV)
.
.
.
.
.
Tektronix Phaser II PXe 17 font: Tektronix Phaser II PXe 17 font (PSCRIPT.DRV)
Tektronix Phaser II PXe 39 font: Tektronix Phaser II PXe 39 font (PSCRIPT.DRV)
Tektronix Phaser II PXi v2010: Tektronix Phaser II PXi v2010 (PSCRIPT.DRV)
Tektronix Phaser III PXi v2010: Tektronix Phaser III PXi v2010 (PSCRIPT.DRV)
Tektronix Phaser IISD v2011: Tektronix Phaser IISD v2011 (PSCRIPT.DRV)
Varityper VT-600: Varityper VT-600 (PSCRIPT.DRV)
Wang LCS15: Wang LCS15 (PSCRIPT.DRV)
Wang LCS15 FontPlus: Wang LCS15 FontPlus (PSCRIPT.DRV)

```

Figure 108. Printer Description List

C.4 Response Files Basics

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. The keywords used in a response file are usually unique to each product.

Response files may include keywords which 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 installation. This allows both installation and configuration to occur concurrently. Configuration keywords may also be used after an installation to modify or reconfigure a currently installed system.

Typically, products will implement some general keywords which are not product-specific. These keywords allow for such things as:

- Including other response files within the one being processed
- Copying files during the install process
- Defining user exits which may be executed during the installation process

These facilities provide flexibility for an administrator who is responsible for defining and implementing a CID installation in a complex environment.

Each product will define its own product-specific keywords and their implementation-specific details. However, there are some general guidelines and rules which all products should abide by.

C.4.1 Response File Processing

The response files will only reside on the CID code server workstations. During the installation process when a response file is found, it uses the installation and configuration information in it as input for the product installation process.

Any erroneous keywords in response files should be ignored by the installation process and the default values for the invalid keywords should be used. If the keyword does not have a default or the configuration is invalid installation will fail. The entire installation should be recorded in a log file.

C.4.2 Group and Client Response Files

When planning and designing a software distribution scenario, an administrator would like to minimize the number of response files which need to be generated and maintained. An administrator would typically like to share the common contents of response files among all applicable users.

This gives the concept of two types of files:

- Group response files
- Client response files

Please refer to the chapters on individual product implementations, or to the product documentation for details on how each product implements these two types of response files. The discussion that follows is general and should apply to most implementations.

A *group response file* will contain the keywords which apply to all users within a specific group. It would generally define the default installation/configuration for those systems that use it, unless specific keywords are overridden by client response files.

```
.  
. Domain=EPSCDEPT  
Sessions=4  
.  
.
```

In the group response file extract previously shown, details that are valid for all or most users have been defined. In this sample, the value for domain name has been supplied along with the number of sessions to be defined for each user.

A *client response file* will usually be used in conjunction with a group response file and will specify only those keywords and values that are unique for the individual client or are different from the values defined in the group response file. In this way a system will be tailored to meet the needs of individual users.

```
.  
. User=John  
Sessions=2  
.  
.
```

In the client response file shown, the user ID has been listed along with the number of sessions that will be required. The number of sessions has been listed because user John does not require four sessions, which had been specified in the group response file. The administrator will want to ensure that the items specified in the client response file will take precedence over those specified in the group response file. For information on ensuring that keywords are processed in the correct sequence, please refer to the product documentation.

C.5 Response File Syntax

This section will address the recommended structure and syntax of response files to be used by CID-enabled products.

A response file is a flat ASCII file that consists of a series of lines that are separated by Carriage Return (x'0D') and/or New Line (x'0A') characters. The syntax used for a response file is straightforward and not restrictive. The response file has a maximum line length of 255 bytes. The lines of a response file fall into two different categories:

- Comment lines
- Response lines

C.5.1 Comment Lines

Comment lines are lines which contain only white-space characters, or have either an asterisk (*), a semicolon (;), or a slash (/) as the first non-white-space character on the line.

White-space characters are defined as tabs, blanks or spaces, and new lines. For example, a line that contains only tab characters followed by a new line sequence should be interpreted as a comment line, as the line contains no characters other than white-space characters. Similarly, if a line only contains a carriage return/line feed sequence, then this line should also be interpreted as a comment line:

```
* This line has an asterisk in column 11
  * This line has an asterisk in column 20
; This line uses a semicolon to indicate a comment

* The line above uses a new line sequence
*** This line is also a valid comment line
```

It should be noted that the asterisk and semicolon only have a special meaning in a response file when they are the first printable character on the line. If the asterisk or semicolon occurs anywhere else on the line, it would be interpreted as part of the string to be assigned to the keyword. The example demonstrates how the asterisk and semicolon only indicate a comment line when they are the first printable character on the line.

```
kwd1 = response * this was intended to be a comment about response
kwd2 = answer ; this was intended to be a comment about answer
```

In the above sample the keywords kwd1 and kwd2 would be assigned everything to the right of the equal sign as their value string. Therefore, comments must always appear as separate lines within a response file.

C.5.2 Response Lines

Response lines are the lines in a response file which are used by the product installation, configuration or maintenance program to determine the options and configuration to install on the target system. Response lines use the following syntax:

```
keyword [= [value]]
```

Where *keyword* is one of a series of string values recognized by the product installation, configuration or maintenance routines and *value*, if present, is the user-assigned value given to that keyword. Note that a value may actually be a list of values. This will be discussed in C.5.4, "Values" on page 468.

C.5.3 Keywords

The *keyword* used in a response line is a string that follows the rules detailed below:

- The keyword begins with an alphanumeric character that is not an asterisk or semicolon.
- It must not contain any imbedded blanks or spaces.
- It does not contain an imbedded or trailing equal (=) sign.
- It should not be case sensitive.

Examples of valid keywords are:

```
include
Include
inCLude
INCLUDE
X1Y23
1abc
A_long_and_drawn_out_but_very_descriptive_keyword
```

In the example above all four *include* examples should equate to the same keyword since response files should not be case sensitive. Following are some examples of keyword combinations that are not valid:

```
=bad_name = 3  
NO Space = y  
Silly=Mistake = 0
```

The first equal sign has been used incorrectly making the first line invalid. In the second line the keyword has been entered with a space included in the name of the keyword. This is not a valid keyword. The third line includes what may be a typing error to include the equal sign in the keyword `Silly_Mistake` instead of the underscore. This would make the keyword invalid. A further complication of this error is that if a keyword called `Silly` existed, then that keyword would be given a value of `Mistake = 0`. Developers should try to ensure that they use unique keyword tokens that would not allow this type of error to occur.

A keyword need not have a value associated with it to be valid in a response file. In some cases, the existence of a keyword can carry significance and there is no additional benefit to assigning it a value. An implementer may choose to define a keyword as having no value associated with it.

C.5.4 Values

In most cases, keywords will have a value (or values) associated with them.

If present, the value associated with a keyword must be preceded by an equal sign (=). It may be separated from the equal sign by zero or more blanks or spaces.

The equal sign itself is a syntax defined constant. It is optional but if present it is separated from the keyword by zero or more blanks or spaces. The equal sign must be placed on the same line as the keyword and, if present, it must be followed by a value also on the same line.

As mentioned above, a keyword may have a single value (or *value_string*) or a *list* of values associated with it.

A *value_string* is an arbitrary string that begins with the first printable character following the equal sign and ends with the last printable character of the line. A *value_string* may not be a one-character string consisting of the left parenthesis ("("). A *value_string* can be a null string. Some valid value strings are:

```
kw1 =  
kw2=john  
kw3 = 8
```

If the value_string consists of a one-character string consisting of the left parenthesis then the assumption will be that this signifies the start of a list. The following example shows an invalid value_string but a value that would be valid as the start of a list:

```
kw4 = (
```

A list of values is a list of response file lines delimited by parentheses. A value is a list when the value_string to the right of the equal sign is a one-character string that consists of the left parenthesis. The equal sign and the left parenthesis character must be on the same line. To avoid any requirement for the use of an escape character, the response file syntax requires that the parentheses be on individual lines.

The syntax used for a list is:

```
keyword = (  
    keyword [= [value]]  
    [keyword [= [value]]]  
)
```

A valid response file showing a list is:

```
kw5 = (  
    kw5.1 = This is  
    kw5.2 = a little  
    kw5.3 = list  
)
```

Lists can also be nested if required.

```
kw6 = (  
    kw6.1 = This is  
    kw6.2 = a response  
    kw6.3 = file  
    kw6.4 = (  
        kw6.4.1 = with a  
        kw6.4.2 = nested list  
    )  
)
```

C.5.5 Including Other Response Files

Response files can be designed so that they can support the inclusion of other response files. (See C.6.2, “Standard Keywords” on page 471 for a detailed discussion of the INCLUDE keyword.) Included response files are files that are read in and processed during the processing of another response file. This provides a means of nesting response files. **A response file should never call itself either directly or indirectly.** Since a response file (including its nested Includes) may contain multiple occurrences of the same keyword or no occurrence of a keyword, it is vital for the user to know in which order the specified keywords will be processed and how they will be interpreted.

C.6 Response File Style Recommendations

There are few limitations imposed on the semantics and processing of a response file. This was intended to provide developers with the most flexibility in designing their programs and response file formats. However, there is a risk that different implementations will result in different interfaces that may be confusing and contradictory. This section will attempt to provide some guidelines, that if followed by all implementers, will provide additional consistency across products.

C.6.1 Keyword Guidelines

Keywords are used to represent actions or a combination of actions and objects.

For example, the keyword INCLUDE represents an action indicating that another response file should be loaded and parsed.

A keyword, SCREEN_SIZE might represent a combination of an action and an object. The action being SET and the object being a screen size parameter.

Keywords that represent both actions and objects are always interpreted in the context of the response file processor. For example, if two products use the same keyword, SESSION_NAME, each product may interpret the keyword and its associated value in a different way. Similarly, a keyword could be interpreted in different ways depending on whether or not it appears in a list in the response file.

To avoid these types of problems and difficulties for the administrator, developers should attempt to ensure that their keyword identifiers are unique.

C.6.2 Standard Keywords

There are certain actions that may be common across many products. When implementing keywords that represent these actions, developers should strive to make the syntax and processing consistent.

IBM has defined three standard functions that are used across several of its CID-enabled products. It is likely that these functions will be useful, if not required by other developers as well. Therefore, this section will address these keywords and suggest guidelines for their implementation.

The standard keywords defined by IBM are as follows:

Keyword	Description
---------	-------------

INCLUDE	Used to include other response files for processing
----------------	---

COPY	Used to copy one file to another
-------------	----------------------------------

USEREXIT	Used to provide a method of performing general user exits
-----------------	---

C.6.2.1 INCLUDE

The INCLUDE keyword accepts a file specification as its value string. The syntax of the include keyword is:

INCLUDE Keyword Syntax
▶▶ INCLUDE =*d:\path\filename.ext*◀◀

where the drive, path and file may be any valid file specification. The file specification used could even be ambiguous because of the inclusion of wildcard characters. If the file specification is ambiguous then only the first file found that matches the file specification criteria should be opened and processed.

When attempting to locate the file, the search for the file should take place in the following sequence:

1. Use the fully qualified file specification if present.
2. Search the current directory for a matching file.
3. Use the filename together with the /G: parameter path if supported by the product.
4. If the file specification is not fully qualified, look on each element of the PATH environment variable.

5. If the file specification is not fully qualified, look on each element of the DPATH environment variable.

The handling code should always process this keyword as soon as it is detected and every time it is detected. The contents of the included response file should be logically imbedded at that point of the including file. When the end of the Include file has been reached, processing should resume at the next line of the response file that contained the Include keyword.

Implementers should log any I/O errors and any failure to find the Include file. If an Include file cannot be located then the implementer should decide whether to abandon processing or attempt to continue from the next line of the Response file.

A Response file should never call itself either directly or indirectly.

Note

The Include keyword is not a valid keyword specification within a value list, that is, a list of keyword-value pairs delimited by parenthesis.

C.6.2.2 COPY

The keyword COPY accepts two file specifications, as expected by the OS/2 copy function, as its value string:

COPY Keyword Syntax

►► COPY == *source target* ◀◀

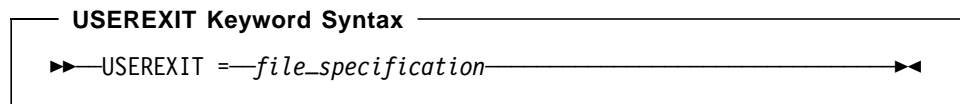
where source is the file specification of the source file to be copied and target is the file specification of the target file.

Each implementation should determine when in the process the COPY is actually carried out and should document it. There are no constraints on developers regarding how they should actually perform the copy.

As most users would expect the COPY to be a plain OS/2 copy, as they are used to when using copy on the OS/2 command line, it should be documented if the COPY actually is doing some sort of 'unpack' and how it is done. It also needs to be documented **when** the COPY is performed. If it is executed when encountered in the response file, first, last or whatever strategy is used.

C.6.2.3 USEREXIT

The USEREXIT keyword accepts a file specification as its value string:



where `file_specification` indicates any valid program file. The file specification could be ambiguous because of the inclusion of wildcard characters. If the file specification is ambiguous then only the first file found that matches the filespec criteria should be opened and processed.

When attempting to locate the file, the search should take place in the following sequence:

1. Use the fully qualified file specification if present.
2. Search the current directory for a matching file.
3. If the file specification is not fully qualified, look on each element of the PATH environment variable.
4. If the file specification is not fully qualified, look on each element of the DPATH environment variable.

Developers should execute the specified command from a CMD.EXE shell in order to ensure that any command files (.CMD) specified are executed correctly.

This keyword is intended for use with general user exits. It is expected that developers will also have specific user exits that will be specified with other keywords. It is up to the individual implementation to determine when a USEREXIT will be processed and to explain this to the user in the product documentation.

C.6.3 Order of Response File Execution

In general, Response file lines should be processed in the same order as they physically appear in the Response file. However, each implementation may have its own requirements, and the developers may decide to process the files in a different order.

Implementers should try to ensure that the Response file processing is done in an order that will be intuitive to the user. Any exceptions to this should be documented.

C.6.4 List Implementation

There is no requirement for lists to be implemented in any particular environment. If a developer chooses to implement lists, they should be used to group data together in a manner that will make things simpler for the user. For example, if there is a function to delete sections of an INI file, then the developer might implement this as shown:

```
INI_delete = (  
    [section1]  
    [section2]  
    [section3]  
)
```

C.7 Response File Processing Sequence

Response files contain sequences of keyword-value pairs which are interpreted during the installation, configuration or maintenance process of a product. Since a response file may contain multiple occurrences of the same keyword or no occurrence of a keyword, it is vital for the user to know in which order the specified keywords will be processed and how they will be interpreted. To facilitate this, a set of rules have been drawn up to assist developers in providing a consistent interpretation and processing structure for keywords that are missing or replicated within a response file.

C.7.1 Response File Hierarchy

There are a number of different scenarios that have to be considered when processing a response file. The developer must decide when to use:

- The default value for a keyword
- A migration value
- A value from an included response file

The set of rules listed here are designed to aid the implementer in making these decisions.

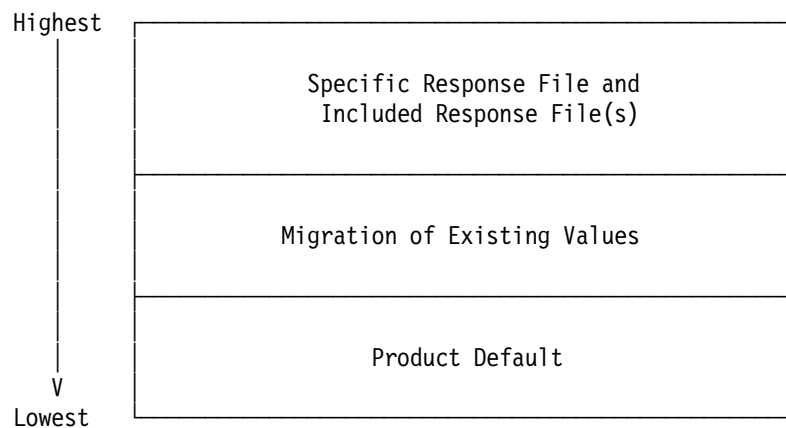
Value	Usage
Product Default	Should be used only when there is no existing value to migrate and no keyword-value exists in any of the referenced response files.
Migration Value	Utilized when there is no keyword value in any referenced response file and the installation process is one that is migrating between product releases. In this

environment, the value from the current configuration is used.

Keyword Value Always use as the value when specified.

Multiple Keyword Instances If multiple instances of the same keyword appear, the last value specified should take precedence.

These rules will help the designer decide which keyword should be used at a particular time. A representation of the hierarchy depicted in these rules is shown below.



If multiple instances of the same keyword appear within the same response file, the last value specified should be used:

```
FORMAT = HPFS
SYSTEM_EDITOR=yes
GAMES = cat_&_mouse solitaire scramble
FORMAT = FAT
PRODUCTIVITY=seek_&_scan epm sticky_pad
```

As we see the same keyword (FORMAT) specified twice, each time with a different value. In this case, the end result will be a FORMAT of type FAT.

C.8 Include Keywords, Detailed Explanation

The following part will explain some keywords in detail and describe how individual response files and results can be managed. The keywords described are:

- Include
- IncludeInLine
- IncludeAtEnd

Different products handle the path of Include files differently.

C.8.1 Single Include and IncludeAtEnd File 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 files 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.

***STANDARD.RSP**


BaseFileSystem=2
DefaultPrinter=50
DisplayAdapter=0

•
•
•

FormatPartition=0
CountryCode=001
Include=X:\RSP\OS2V211\INDIV.RSP

•
•
•

Mouse=1
PrinterPort=1
REXX=0



***INDIV.RSP**
FormatPartition=1
BaseFileSystem=1
REXX=1
DisplayAdapter=5

Note

The entire STANDARD.RSP response file will be processed from beginning to end, and only then the include file, namely INDIV.RSP, will 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 REXX installed on it.

IncludeAtEnd: Works as the Include keyword described above.

C.8.2 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 files 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.

***STANDARD.RSP**

BaseFileSystem=2
DefaultPrinter=50
DisplayAdapter=0

•
•
•

FormatPartition=0
CountryCode=001
IncludeInLine=X:\RSP\OS2V211\INDIV.RSP

•
•
•

Mouse=1
PrinterPort=1
REXX=0

***INDIV.RSP**
FormatPartition=1
BaseFileSystem=1
REXX=1
DisplayAdapter=5

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 this example REXX=0 is found in the STANDARD.RSP after the IncludeInLine keyword, this will override REXX=1 specified in INDIV.RSP since REXX=0 is the last value found for the REXX keyword.

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 REXX will not be installed on it because REXX=0 found in STANDARD.RSP overrides REXX=1 specified in INDIV.RSP.

C.8.3 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 having one ConfigSysLine with a remark telling which MOUSE(X).INC was called and the appropriate mouse function number.

The content of file MOUSE1.INC is shown below:

```
ConfigSysLine=REM mouse1 include passed  
mouse=1
```

The MOUSE2.INC has one ConfigSysLine with a remark plus 2 Include statements. The content of file MOUSE45.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, etc.). 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=2
Processing: ConfigSysLine=REM mouse4 include passed
Processing: mouse=3
Processing: ConfigSysLine=REM mouse5 include passed
Processing: mouse=4
      •
      •
```

The following figure visualizes the process explained above.

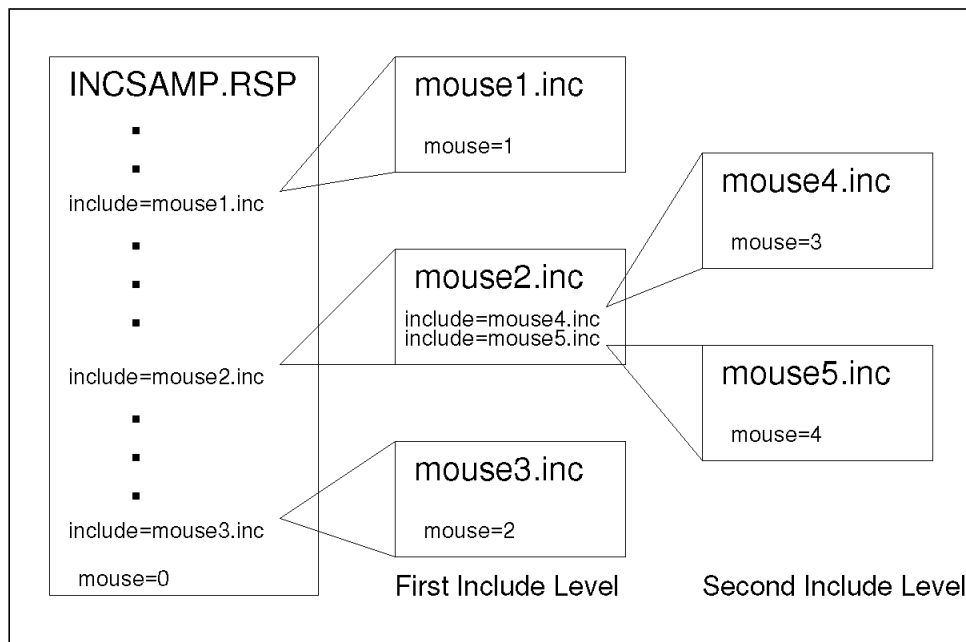


Figure 109. Response File Multiple Include Pattern

C.9 The User Exit Keywords of the Response File

The response file has two exit keywords:

- EarlyUserExit and
- 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 will use 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, then 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.CMD 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.CMD) 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.

Appendix D. OS/2 V2.1 CID Installation Utility for SVGA Adapters

The utilities for remote installation and configuration of SVGA adapters can be found in the SVGACID subdirectory of the sample code CDROM. They support **only** OS/2 V2.1. We supply these utilities "as is" without any warranty or support by IBM.

The utilities have to be invoked after the initial install of the client workstation, which means after at least one reboot has been executed from the hard disk. OS/2 V2.1 has to be installed, including DOS support. During the initial install, VGA, SVGA or AutoDetection should be specified in the response file, using the DisplayAdapter keyword.

The SVGA adapters must be supported by OS/2 V2.1 and have at least 1MB of video RAM defined in the adapter settings.

Please note that the install of the OS/2 V2.1 Service Pak XR06200 may overwrite the display driver created by SVGACID. You may want to consult the 211DUU Package - Display Driver Update Utility - if you plan to apply the Service Pak.

D.1 Usage of SVGA CID Support

You have to follow these steps to use the utilities:

1. Preparation of the code server

A directory has to be created to hold the necessary files. Create the directory following this example; replace **D:** with your drive containing the CID directory structure:

```
MD D:CIDIMSGVACID
```

Copy all files from the \SVGACID of the sample code CDROM into this subdirectory.

```
COPY G:SVGACID D:CIDIMSGVACID
```

2. Changes in the LCU command file

The product definitions for the utilities are already included in the default LCU command file that is provided on the sample code CDROM. The following figure shows them:

```

x.video = 45                                /* structure index */
x.45.name='CID SVGA Install for OS/2 2.1'    /* product name */
x.45.statevar = 'CAS_' || x.45.name          /* state variable name */
x.45.instprog = 'x:\img\svgacid\install.cmd', /* fully qualified install program name */
          'x:\img\svgacid' || bootdrive || ' /h' /* source, target, resolution */
x.45.rspdir = ''                             /* response file directory */
x.45.default = ''                           /* default response file */

x.dspinstl = 46                             /* structure index */
x.46.name='SVGA Dspinstl'                   /* product name */
x.46.statevar = 'CAS_' || x.46.name          /* state variable name */
x.46.instprog = bootdrive || '\os2\install\dspinstl.exe', /* fully qualified install program name */
          '/pk:SVGA',                        /* Primary key for display adapter */
          '/sk:NONE',                        /* Secondary key for display adapter */
          '/s:x:\img\os2v21',                /* - source directory */
          '/t:' || bootdrive,                /* - target drive */
          '/mc:3'                            /* - manufacturer code */
x.46.rspdir = ''                             /* response file directory */
x.46.default = ''                           /* default response file */

x.restore = 47                              /* structure index */
x.47.name='Restore client DSC files'         /* product name */
x.47.statevar = 'CAS_' || x.47.name          /* state variable name */
x.47.instprog = 'x:\img\svgacid\restore.cmd', /* fully qualified install program name */
          '' || bootdrive                    /* target drive */
x.47.rspdir = ''                             /* response file directory */
x.47.default = ''                           /* default response file */

```

Figure 110. Product Definition Part for the SVGA Utilities

The installation part of the LCU command file has to be modified:

```

Do Forever
  Select
    when OVERALL_STATE = 0 then do
      if BootDrive() == 'DISKETTE' then iterate /* Check if booted from diskette*/
                                              /* if it was, then goto state 1*/
      if RunInstall(x.semaint210) == BAD_RC then exit /* Install maintenance system */
      if RunInstall(x.laps_prep) == BAD_RC then exit /* Install LAPS prep system */
      if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 1 then do
      if RunInstall(x.seinst210) == BAD_RC then exit /* Install operating system */
      if RunInstall(x.laps) == BAD_RC then exit /* Install LAPS */
      if RunInstall(x.thinifs1) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.thinifs2) == BAD_RC then exit /* Install SRVIFS requester */
      if RunInstall(x.casinst1) == BAD_RC then exit /* Install LCU */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 2 then do
      if RunInstall(x.video) == BAD_RC then exit /* Prepare client for SVGA */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 3 then do
      if RunInstall(x.dspinst1) == BAD_RC then exit /* Run DSPINSTL for SVGA */
      Call CheckBoot /* Reboot if it was requested */
    end
    when OVERALL_STATE = 4 then do
      if RunInstall(x.restore) == BAD_RC then exit /* Restore Client files */
      Call RebootAndGotoState(5) /* Reboot */
    end
    when OVERALL_STATE = 5 then do
      if RunInstall(x.ifsdel) == BAD_RC then exit /* Delete THINIFS */
      if RunInstall(x.casdelet) == BAD_RC then exit /* Delete LCU */
      Call Reboot /* Reboot */
    end
  end
end
exit

```

Figure 111. Installation Part for the SVGA Utilities

D.2 Detailed Description of the Utilities

- INSTALL.CMD

The INSTALL.CMD copies the SVGACID.DLL to the client workstation and saves the clients' *.DSC files. It unpacks new *.DSC files according to the resolution that was specified in the invocation. It is invoked with the parameters [SOURCE DRIVE:] [TARGET DRIVE:] [RESOLUTION]

where:

<SOURCE DRIVE:> - path to CID SVGA files on this diskette.

<TARGET DRIVE:> - target drive on client workstation.

<RESOLUTION> is one of the following:

/h,/H,-h,-H for 1024x768x256 high resolution

/m,/M,-m,-M for 800x600x256 medium resolution

/l,/L,-l,-L for 640x480x256 low resolution

- **DSPINSTL.EXE**

DSPINSTL.EXE is an OS/2 executable that is used to install display drivers. It can be invoked without parameters and will then guide the user with a PM interface or it can be invoked with parameters and executes then without user prompting. It resides in the OS2INSTALL subdirectory and is invoked from there.

It is invoked with the following parameters:

/PK: primary adapter key

/SK: secondary adapter key

/S: source drive

/T: target drive

/MC: manufacturing code of supported OS/2 2.1 SVGA adapters

Invocation example:

DSPINSTL.EXE /PK:SVGA /SK:NONE /S:X:IMG0S2V21 /T:C: /MC:3

The product definition part for DSPINSTL in the LCU command file has to be changed to reflect your hardware.

To find out which SVGA adapter you have in a PS/ValuePoint, refer to Chapter 4, Video Subsystem, in *&vpbook.*, *&vpbooki.*. The following figure shows the manufacturer codes that are related to the OS/2 V2.1 supported SVGA adapters.

<i>Table 18 (Page 1 of 2). Overview of Manufacturing Codes for SVGA Adapters</i>	
Adapter	Manufacturing Code
Headland	1
Trident	2
Tseng ET4000	3
Western Digital	4

<i>Table 18 (Page 2 of 2). Overview of Manufacturing Codes for SVGA Adapters</i>	
Adapter	Manufacturing Code
ATI	5
Speedway	6
Cirrus Logic	7

- **RESTORE.CMD**

The RESTORE.CMD deletes the SVGACID.DLL from the client workstation. and restores the original OS/2 V2.1 display configuration files. It is invoked with the parameter:

<TARGET DRIVE:> - target drive on client workstation.

After this procedure, the client workstation has to be rebooted. In the LCU command file, this is accomplished by the procedure call RebootAndGotoState(x) because none of the procedures automatically calls for a reboot.

NVDM/2 Usage

If you are using NVDM/2 as the software distribution manager, you have to create change file profiles for the utilities that follow the description given above. You can execute the three procedures in a *coreq group* giving the RESTORE.CMD a *PhaseEnd=Yes* to force a reboot.

Appendix E. LAN Network Adapters

The following table contains network adapter driver descriptions, the device driver file name and the associated NIF file name for the network adapter drivers. File creation dates and file lengths are also noted where available.

The Network Information File (NIF) contains network adapter information, such as adapter name(s), name of network adapter driver, valid value ranges and so on and is used when LAPS is installed. A NIF file should usually not be changed. The NIF file name is given as a parameter when configuring LAN transport via response files and also with the THINLAPS utility.

E.1 NDIS 2.0.1 MAC Drivers

Additional information is provided in text files for some adapter cards. The text file usually has the name xx.txt, where xx is the same as the device driver for that adapter card. The text file is listed in the table below.

When a file is delivered both in MPTS and by its manufacturer on a driver diskette, and the file names differ, then the file name given by the manufacturer is noted in parenthesis.

Table 19 (Page 1 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
3Com Corporation			
3Com EtherLink II (3C503) Note: EtherDisk II diskette from 3Com.	ISA	ELNKII.OS2 8-6-91 17902 ELNKII.NIF (LS503OS2.NIF) 2-5-93 3030	ELNKII.DOS 8-6-91 11322 LS503DOS.NIF 2-5-93 3029
3Com EtherLink II-16 (3C503-16)	ISA	ELNKII.OS2 8-6-91 17902 ELNKII.NIF (LS503OS2.NIF) 2-5-93 3030	ELNKII.DOS 8-6-91 11322 LS503DOS.NIF 2-5-93 3029
3Com EtherLink II-16-TP (3C503-16-TP)	ISA	ELNKII.OS2 8-6-91 17902 ELNKII.NIF (LS503OS2.NIF) 2-5-93 3030	ELNKII.DOS 8-6-91 11322 LS503DOS.NIF 2-5-93 3029
3Com EtherLink 16 (3C507)	ISA	ELNK16.OS2 3-10-93 10540 LS507OS2.NIF 2-17-93 957	ELNK16.DOS 11-13-92 9800 LS507DOS.NIF 2-17-93 956
3Com EtherLink 16-TP (3C507-TP)	ISA	ELNK16.OS2 3-10-93 10540 LS507OS2.NIF 2-17-93 957	ELNK16.DOS 11-13-92 9800 LS507DOS.NIF 2-17-93 956

<i>Table 19 (Page 2 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
3Com EtherLink/MC (3C523B)	MCA	ELNKM.C.OS2 8-5-92 16608 ELNKM.C.NIF (LS523OS2.NIF) 3-1-93 1509	ELNKM.C.DOS 08-05-92 9542 LS523DOS.NIF 10-04-94 1506
3Com EtherLink/MC-TP (3C523B-TP)	MCA	ELNKM.C.OS2 8-5-92 16608 ELNKM.C.NIF (LS523OS2.NIF) 3-1-93 1509	ELNKM.C.DOS 08-05-92 9542 LS523DOS.NIF 10-04-94 1506
3Com EtherLink MC-32 (3C527B)	MCA busmaster	ELMC32.OS2 5-2-91 34109 LS527OS2.NIF 4-28-93 1432	ELMC32.DOS 05-15-91 8926 LS527DOS.NIF 04-28-93 1432

Table 19 (Page 3 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
<p>3Com EtherLink III (3C509)</p> <p>Note: For all EtherLink III:</p> <ul style="list-style-type: none"> • Use the EtherDisk III diskette from 3Com, current version 4.2; • Be sure to run 3Com configuration and set to OS/2 mode vs. DOS mode. • protocol.ini parameters: <ul style="list-style-type: none"> – DRIVERTNAME=ELNK3\$ – ;DRIVERTNAME=ELNK32\$ – ; use for second of two MAC drivers – ;IOADDRESS=0X300 – ; only ISA with > 1 NIC – ; range 0X200-0X3E0, Step 0X10 – MAXTRANSMITS=40 – ; range 2-50, use 40 for OS/2 Servers – ;NETADDRESS="000000000000" – ;SLOT=0 – ; range 0-15, use for EISA only <p>Note: An LT80209 error can occur during IPL if the 3Com 3C509 is configured for a maximum modem speed of 38,400 and IBM's Netware Requester Support (odi2ndi.os2) is also configured. Lowering the maximum speed to 19,200 allows the system to IPL without the error.</p> <p>Note: For B-series cards (3C509B-Combo): the adapter is software configured using the 3C5X9CFG.EXE utility found on the EtherDisk III V. 4.2. It configures the following parameters:</p> <ul style="list-style-type: none"> • I/O Base Address (default=300h) • Interrupt Request Level (default=10) • Boot PROM (default=disabled) • Transceiver Type (default=Auto Select) • Network Driver Optimization (default=Windows or OS/2 Client) • Maximum Modem Speed (default=9600 Baud) • Plug and Play Compatibility (default=Enabled) <p>Note: For B-Series cards, Plug and Play must be disabled.</p>	ISA	<p>ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043</p>	<p>ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038</p>
3Com EtherLink III (3C509B)	ISA	<p>ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043</p>	<p>ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038</p>
3Com EtherLink III-Combo (3C509-COMBO)	ISA	<p>ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043</p>	<p>ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038</p>

<i>Table 19 (Page 4 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
3Com EtherLink III-Combo (3C509B-COMBO)	ISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-TP (3C509-TP)	ISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-TP (3C509B-TP)	ISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-TPO (3C509-TPO)	ISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-TPO (3C509B-TPO)	ISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-EISA (3C579)	EISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-EISA-TP (3C579-TP)	EISA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-MCA (3C529)	MCA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-MCA-TP (3C529-TP)	MCA	ELNK3.OS2 07-29-94 25147 EL3IBMO2.NIF (LS5X9OS2.NIF) 07-18-94 1043	ELNK3.DOS 07-29-94 15519 EL3IBMDS.NIF 07-18-94 1038
3Com EtherLink III-PCMCIA-TP (3C589-TP)	PCMCIA	ELPC3.OS2 4-20-95 29983 ELPC3OS2.NIF 8-2-94 1097	
3Com EtherLink III-PCMCIA-Combo (3C589B-Combo)	PCMCIA	ELPC3.OS2 4-20-95 29983 ELPC3OS2.NIF 8-2-94 1097	
3Com EtherLink III LAN PC Card (3C589C)	PCMCIA	ELPC3.OS2 8-11-95 29983 ELPC3OS2.NIF 8-17-95 1615	
3Com EtherLink III LAN+Modem PC Card (3C562)	PCMCIA		
3Com EtherLink III PCI 10BASE-T Network Adapter (3C590-TPO)	PCI busmaster	EL59X.OS2 09-18-95 27787 LS59XOS2.NIF 05-03-95 1357	

<i>Table 19 (Page 5 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
3Com Fast EtherLink PCI 10/100 BASE-T Network Adapter (3C595-TX)	PCI	EL59X.OS2 09-18-95 27787 LS59XOS2.NIF 05-03-95 1357	
3Com Fast EtherLink EISA 10/100 BASE-T Network Adapter (3C597-TX)	EISA	EL59X.OS2 09-18-95 27787 LS59XOS2.NIF 05-03-95 1357	
3Com TokenLink III (3C619) Note: TokenDisk V. 1.1 diskette from 3Com. Note: Do not use the OS/2 NDIS drivers on the TokenDisk.	ISA	IBMTOK.OS2 TLNKIII.NIF LT2.MSG LT2H.MSG	
3Com TokenLink III EISA (3C679)	EISA	IBMTOK.OS2 TLNKIII.NIF LT2.MSG LT2H.MSG	
3Com TokenLink III MCA (3C629)	MCA	IBMTOK.OS2 TLNKIII.NIF LT2.MSG LT2H.MSG	
3Com TokenLink III 16/4 PC Card (3C689)	PCMCIA	TLCP3.OS2 03-13-95 25404 TLCP3.NIF 02-01-95 1051	
Accton			
Accton EtherCombo-16 (EN1650)	ISA	ETHNE.OS2 EN165X0.NIF	
Accton EtherPair-16 (EN1651)	ISA	ETHNE.OS2 EN165X0.NIF	
Accton EtherCoax-16 (EN1652)	ISA	ETHNE.OS2 EN165X0.NIF	
Accton EtherCombo-32 (EN1200)	EISA	ACC1200.OS2 ACC1200E.NIF	
Advanced Micro Devices, Inc.			
AMD PCnet-32 Ethernet Adapter	VESA	PCNTND.OS2 02-21-95 80298 PCNTND.NIF 11-21-93 207	
AMD PCnet-ISA II Ethernet Adapter	ISA	PCNTND.OS2 02-21-95 80298 PCNTND.NIF 11-21-93 207	
AMD PCnet-PCI Ethernet Adapter	PCI	PCNTND.OS2 02-21-95 80298 PCNTND.NIF 11-21-93 207	
Allied Telesis			
Allied Telesis Ethernet Adapter Card ISA (AT1500-Plus)	ISA	AT1500.OS2 2-2-94 11283	
Allied Telesis AT1700 Plus ISA	ISA	AT1700.OS2 1-28-94 7448	

<i>Table 19 (Page 6 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Allied Telesis AT1720 Plus MCA	MCA	AT1700.OS2 1-28-94 7448	
Artisoft			
Artisoft NodeRunner/SI 2000/C Note: These Artisoft NICs might be detected as NE1000, NE2000, or Artisoft AE-2, AE-3 NICs. Configure to use the NodeRunner driver.	ISA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft NodeRunner/SI 2000/T	ISA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft NodeRunner/SI 2000/A	ISA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft NodeRunner/SI 2000M/TC	MCA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft LANTastic NodeRunner 2000/C	ISA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft LANTastic NodeRunner 2000/T	ISA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft LANTastic NodeRunner 2000/A	ISA	AEXNDIS.OS2 AEXNDIS.NIF	
Artisoft LANTastic NodeRunner 2000M/TC	MCA	AEXNDIS.OS2 AEXNDIS.NIF	
Asante Note: The current NIF from Asante is not compatible with LAPS. The user must create a NIF manually. See the detailed instructions in READMAC.TXT.			
Asante EtherPaC 2000+3	ISA	EP2000.OS2 7-13-94 EP2000.NIF	
Asante EtherPaC 2000+N	ISA	EP2000.OS2 7-13-94 EP2000.NIF	
Asante EtherPaC 2000+T	ISA	EP2000.OS2 7-13-94 EP2000.NIF	
Cabletron Corporation			
Cabletron Ethernet DNI Adapter (E1112)	ISA	E11ND.OS2 E11.NIF	
Cabletron Ethernet DNI Adapter (E1119)	ISA	E11ND.OS2 E11.NIF	
Cabletron Ethernet DNI Adapter (E2112)	ISA	E21ND.OS2 E21.NIF	
Cabletron Ethernet DNI Adapter (E2119)	ISA	E21ND.OS2 E21.NIF	
Cabletron Ethernet DNI Adapter (E3112)	MCA	E31ND.OS2 E31.NIF	
Cabletron Ethernet DNI Adapter (E3119)	MCA	E31ND.OS2 E31.NIF	
Cabletron Token-Ring DNI Adapter (T2015)	ISA	T20ND.OS2 T20.NIF	
Cabletron Token-Ring DNI Adapter (T3015)	MCA	T30ND.OS2 T30.NIF	
CeLAN			
CeLAN FlexLINK - EPCiplus (9910EPCI-B)	PCI	PCIND.OS2 1-4-95 15790 RTL8029.NIF 1-22-95 3394	

Table 19 (Page 7 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Cogent Data Technologies, Inc.			
Cogent eMASTER+ EM960 PCI Ethernet Adapter (EM960C)	PCI	EMPCI.OS2 3-21-95 18121 EMPCI.NIF 3-22-95 8334	
Cogent EM100 PCI Fast Ethernet Adapter	PCI	EMPCI.OS2 3-21-95 18121 EMPCI.NIF 3-22-95 8334	
Compaq			
Compaq NetFlex-2 ENET-TR Controller Note: This card supports ethernet. With an upgrade chip, it also supports token-ring.	EISA busmaster	NETFLX.OS2 V 1.11 5-19-94 NETFLX.NIF	
Cray Communications Note: Formerly called Dowty Networks			
Cray Communications ScaNet Network Interface Adapter-ISA	ISA	PC04.OS2 PC04.NIF PC04CRAY.NIF	
Cray Communications ScaNet Network Interface Adapter-MCA	MCA	PC04.OS2 PC04.NIF PC04CRAY.NIF	
Digital Communications Associates			
DCA ClassicBlue MC 4/16 Token-Ring Adapter	MCA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
DCA IRMATrac EISA	EISA	OLITOK32.OS2 2-16-94 46888 OLITOK32.NIF 2-15-94 4180	
DCA IRMATrac Token-Ring Adapter/Convertible-MCA	MCA	IRMATR.OS2 IRMATR.NIF IRMATR.TXT	
Digital Equipment Corporation			
Digital EtherWorks 3 Turbo TP (DE204-AA)	ISA		
Digital EtherWorks Turbo 435 PCI	PCI	DC21040.OS2 6-29-94 10853 DC21040.NIF 10-5-94 951	
Digital Semiconductor			
EB40-DECchip 21040 evaluation board	PCI	DC21X4.OS2 08-03-95 16069 DC21X4.NIF 08-03-95 1001	
EB140-DECchip 21140 evaluation board	PCI	DC21X4.OS2 08-03-95 16069 DC21X4.NIF 08-03-95 1001	
D-Link			

<i>Table 19 (Page 8 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
D-Link Ethernet Interface Card for the PC XT/AT (DE-220C) Note: For all D-Link DE-220 family adapter cards, the DE-220 device driver (DE220.OS2) currently provided on the driver diskette does not function properly with LAPS (cannot logon). The user must obtain the current DE-200 driver from the BBS. This driver is compatible with the DE-220 adapter cards. The user must also obtain an alternate NIF. See the detailed instructions in READMAC.TXT. Note: The DE-220 driver diskette may have been revised to include a working driver. I am investigating. 07/15/95.	ISA	DE200.OS2 DE200IBM.NIF	
D-Link Ethernet Interface Card for the PC XT/AT (DE-220CAT)	ISA	DE200.OS2 DE200IBM.NIF	
D-Link Ethernet Interface Card for the PC XT/AT (DE-220CT)	ISA	DE200.OS2 DE200IBM.NIF	
D-Link Ethernet Interface Card for the PC XT/AT (DE-220T)	ISA	DE200.OS2 DE200IBM.NIF	
D-Link Ethernet Card for EISA bus PC (DE-400)	EISA	DE400.OS2 DE400.NIF	
D-Link Ethernet VESA Combo Card (DE-500CAT)	VESA	DE500.OS2 DE500IBM.NIF	
D-Link Ethernet PCI (DE-530CT)	PCI	DC21X4.OS2 6-9-95 14196 DC21X4I.NIF 6-7-95 2041	
D-Link Token-Ring Adapter for the PC/AT and PS/2 (DT-220)	ISA		
Eagle Technology Note: The user must set the timing parameter on the NE2000 family to a non-default position on some systems, including some IBM PS Valuepoint systems. This is set via a jumper on the older NE2000 boards, and via software configuration on the newer NE2000PLUS boards. See the instructions in the Eagle manual.			
Eagle Novell NE2000	ISA	NE2000.OS2 IBMNE200.NIF	
Eagle Novell NE2000T	ISA	NE2000.OS2 IBMNE200.NIF	
Eagle Novell NE2000plus	ISA	NE2000.OS2 IBMNE200.NIF	
Eagle Novell NE2000Tplus	ISA	NE2000.OS2 IBMNE200.NIF	
Eagle Novell NE2000plus-3	ISA	NE2000.OS2 IBMNE200.NIF	
Eagle EtherXpert EP2000plus	ISA	EP2000.OS2 IBMNEP200.NIF	
Eagle EtherXpert EP2000Tplus	ISA	EP2000.OS2 IBMNEP200.NIF	

<i>Table 19 (Page 9 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Eagle Novell NE3210	EISA	NE3210.OS2 IBMNE321.NIF	
Eagle EtherXpert EP3210	EISA	EP3210.OS2 IBMEP321.NIF	
Eagle Novell NE/2T	MCA	N/A	
Hewlett-Packard			
Hewlett-Packard 27247B	ISA	HPLANP.OS2 10-12-93 HPLANP.NIF 9-17-93	
Hewlett-Packard PCLAN Adapter/16 PLUS (27252A)	ISA	HPLANP.OS2 12-02-94 13706 HPLANP.NIF 11-22-94 3674	
Hewlett-Packard JP2405A	ISA		
Hewlett-Packard 10/100VG PC LAN ISA Adapter (J2573A)		HPFEND.OS2 10-03-94 13840 HPFEND.NIF 10-25-94 1984	
Hewlett-Packard 10/100VG PC LAN EISA Adapter (J2577A)	EISA	HPFEND.OS2 05-15-95 18560 HPFEND.NIF 05-15-95 4634	
Hewlett-Packard 10/100VG PC LAN PCI Adapter (J2585A)	PCI	HPFEND.OS2 04-04-95 18576 HPFEND.NIF 06-06-94 4713	
IBM Corporation Note: IBMTOKC.NIF is provided as a general NIF for IBM shared-RAM token-ring NICs. This is selected as "IBM Compatible Token-Ring Network Adapter" in the LAPS panel. This is selected automatically for both IBM and non-IBM NICs during adapter auto-detection. This is equivalent to selecting IBMTOK.NIF. Note: IBMTOKMP.OS2, IBMTOKMP.NIF, and IBMTOKMP.TXT are provided for use on SMP systems in place of IBMTOK.OS2. This is supported only on SMP systems. On SMP systems, the user should select "IBM SMP Token-Ring Network Adapter" for both IBM and non-IBM adapters which are supported on SMP. All other NICs which are supported on SMP systems use the same device driver that is used on uni-processor systems.			
IBM LAN Adapter for Ethernet (48G7169)	ISA	IBMENI.OS2 IBMENI.NIF IBMENI.TXT	
IBM LAN Adapter for Ethernet CX (60G615)	ISA	IBMENI.OS2 IBMENI.NIF IBMENI.TXT	
IBM LAN Adapter for Ethernet TP (60G0605)	ISA	IBMENI.OS2 IBMENI.NIF IBMENI.TXT	
IBM EtherJet ISA Adapter	ISA	IBMEINDI.OS2 10-28-95 37654 IBMEINDI.NIF 10-21-95 1028	

<i>Table 19 (Page 10 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM EtherJet 10-Base-T ISA Adapter	ISA	IBMEINDI.OS2 10-28-95 37654 IBMEINDI.NIF 10-21-95 1028	
IBM Ethernet Network Adapter 10BaseT 66G0939 (JAPAN ONLY)	ISA		
IBM Ethernet Network Adapter 10Base2 66G0943 (JAPAN ONLY)	ISA		
IBM Credit Card Adapter for Ethernet 10B2 (0933280)	PCMCIA	PCMNICCS.OS2 PCMNICCS.NIF PCMNICCS.TXT	
IBM Credit Card Adapter for Ethernet 10BT (0933290)	PCMCIA	PCMNICCS.OS2 PCMNICCS.NIF PCMNICCS.TXT	
IBM Credit Card Adapter II for Ethernet 10B2 (0934330)	PCMCIA	PCMNICCS.OS2 PCMNICCS.NIF PCMNICCS.TXT	
IBM Credit Card Adapter II for Ethernet 10BT (0934331)	PCMCIA	PCMNICCS.OS2 PCMNICCS.NIF PCMNICCS.TXT	
IBM Adapter/A for Ethernet Networks (6451091)	MCA	MACETH.OS2 MACETH2.OS2 MACETH.NIF ETH.MSG ETHH.MSG	
IBM Adapter/A for Ethernet Twisted-Pair Networks (6451136)	MCA	MACETH.OS2 MACETH2.OS2 MACETH.NIF ETH.MSG ETHH.MSG	
IBM Ethernet Network Adapter/A 10Base2/5 35G2793 (JAPAN ONLY)	MCA	MACETH.OS2 MACETH2.OS2 MACETH.NIF ETH.MSG ETHH.MSG	
IBM Ethernet Network Adapter/A 10Base5/T 35G2806 (JAPAN ONLY)	MCA	MACETH.OS2 MACETH2.OS2 MACETH.NIF ETH.MSG ETHH.MSG	
IBM LAN Adapter/A for Ethernet (48G7171)	MCA	IBMENII.OS2 IBMENII.NIF IBMENII.TXT	
IBM EtherStreamer MC 32 Adapter (59G9066)	MCA busmaster	IBMMPC.OS2 V 1.3 or higher IBMMPC.NIF LTC.MSG LTCH.MSG IBMMPC.TXT	

<i>Table 19 (Page 11 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM EtherStreamer MC 32 Adapter (74G0883) (JAPAN ONLY)	MCA busmaster	IBMMPC.OS2 12-18-93 42036 IBMMPC.NIF 06-16-93 6881 LTC.MSG 06-16-93 3276 LTCH.MSG 06-16-93 10614 IBMMPC.DOC 02-11-94 2745	IBMMPC.DOS
IBM Dual EtherStreamer MC 32 Adapter (73G7136) Note: Add DEVICE= ...\IBMCOM\MACS\DUALSTRM.OS2 when using both ports. Configure one logical adapter for each port required. I.e, select IBMMPC twice in LAPS to configure both ports of a single DUAL Streamer NIC.	MCA busmaster	IBMMPC.OS2 V 3.0 or higher IBMMPC.NIF DUALSTRM.OS2 LTC.MSG LTCH.MSG IBMMPC.TXT	
IBM Ethernet Quad-BT PeerMaster Server Adapter (06H5184)	MCA	MXMCA4BT.OS2 12-5-94 8248 MXMCA4TN.OS2 12-5-94 8248 VNET.OS2 12-8-94 5246 MXMCA4BT.NIF 11-16-94 3429 MXMCA4TN.NIF 12-5-94 3429 VNET.NIF 12-8-94 5246 MXMCA4BT.BIN 12-4-94 89174 MXMCA4TN.BIN 12-4-94 89126	
IBM Ethernet Quad-B2 PeerMaster Server Adapter (06H6041)	MCA	MXMCA4BT.OS2 12-5-94 8248 MXMCA4TN.OS2 12-5-94 8248 VNET.OS2 12-8-94 5246 MXMCA4BT.NIF 11-16-94 3429 MXMCA4TN.NIF 12-5-94 3429 VNET.NIF 12-8-94 5246 MXMCA4BT.BIN 12-4-94 89174 MXMCA4TN.BIN 12-4-94 89126	
IBM Token-Ring PC Network Adapter	ISA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	

<i>Table 19 (Page 12 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM Token-Ring PC Network Adapter II	ISA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM Token-Ring Network 16/4 Adapter	ISA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM Token-Ring Network 16/4 ISA-16 Adapter (73G2032)	ISA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM Token-Ring Network 16/4 Adapter II	ISA busmaster 24-bit DMA	IBM16TR.OS2 IBM16OS2.NIF BRZ.MSG BRZH.MSG IBM16TR.TXT	
IBM Auto 16/4 Token-Ring ISA Adapter (92G7632) Note: LS 4.0: Recommend the new driver (v. 2.04) or later. Note: Warp Connect: Contains current driver V. 2.6. Note: Cannot connect the TP cable to a DTAU requiring power.	ISA	IBMTOK.OS2 9-10-94 16296 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM 16/4 Busmaster EISA Adapter (1051712)	EISA busmaster	IBMEITR.OS2 IBMEIOS2.NIF LT9.MSG LT9H.MSG	
IBM Token-Ring 16/4 Credit Card Adapter (0933462)	PCMCIA	IBMTOKCS.OS2 IBMTOKCS.NIF LTG.MSG LTGH.MSG IBMTOKCS.TXT	
IBM Token-Ring 16/4 Credit Card Adapter II (0933931)	PCMCIA	IBMTOKCS.OS2 IBMTOKCS.NIF LTG.MSG LTGH.MSG IBMTOKCS.TXT	
IBM PCMCIA Token-Ring Adapter (04H6922) Note: on the Thinkpad 701, the user should change MMIO from CC00 to another address such as D400.	PCMCIA	IBMTOKCS.OS2 IBMTOKCS.NIF LTG.MSG LTGH.MSG IBMTOKCS.TXT	
IBM Token-Ring Network Adapter/A (69X8138)	MCA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM Token-Ring Network 16/4 Adapter/A (16F1163)	MCA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM Token-Ring Network 16/4 Adapter/A (74F9410)	MCA	IBMTOK.OS2 IBMTOK.NIF LT2.MSG LT2H.MSG	

<i>Table 19 (Page 13 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM Auto 16/4 Token-Ring MC Adapter (92G7682) Note: LS 4.0: Recommend the new driver (v. 2.6) or later. Note: Warp Connect: Contains current driver V. 2.6.	MCA	IBMTOK.OS2 3-23-95 17736 IBMTOK.NIF LT2.MSG LT2H.MSG	
IBM 16/4 Busmaster Server Adapter/A (74F4140) Note: The files MONT400.BIN and WRTRAM.BIN must be in the \IBMCOM\MACS directory.	MCA busmaster 24-bit dma	IBMTRBM.OS2 IBMTRBM.NIF MONT400.BIN WRTRAM.BIN LT4.MSG LT4H.MSG	
IBM LANStreamer MC 16 Adapter (74G0801)	MCA busmaster 24-bit dma	IBMTRDB.OS2 IBMTRDB.NIF LT6.MSG LT6H.MSG IBMTRDB.TXT	
IBM LANStreamer MC 32 Adapter (74G0103)	MCA busmaster	IBMTRDB.OS2 IBMTRDB.NIF LT6.MSG LT6H.MSG IBMTRDB.TXT	
IBM Auto LANStreamer MC 32 Adapter (60G1592)	MCA busmaster	IBMMPC.OS2 V 2.0 or higher IBMMPC.NIF LTC.MSG LTCH.MSG IBMMPC.TXT	
IBM Dual LANStreamer MC 32 Adapter (73G7137) Note: Add DEVICE=...\\IBMCOM\MACS\DUALSTRM.OS2 when using both ports. Configure one logical adapter for each port required. I.e, select IBMMPC twice in LAPS to configure both ports of a single DUAL Streamer NIC.	MCA busmaster	IBMMPC.OS2 V 3.0 or higher IBMMPC.NIF DUALSTRM.OS2 LTC.MSG LTCH.MSG IBMMPC.TXT	
IBM Auto LANStreamer PCI Adapter (04H8095) Note: Some systems require BIOS upgrade and OS/2 DMP or DASD driver to fix PCI conflict.	PCI busmaster	IBMMPC.OS2 V 4.01.00 or higher 05-22-95 61492 IBMMPC.NIF 05-22-95 9645 LTC.MSG LTCH.MSG IBMMPC.TXT	
IBM FDDI Copper Base ISA Adapter	ISA	IBMFDX.OS2 5-11-94 101744 IBMFDX.NIF 5-18-94 13226 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM FDDI Copper Base MCA Adapter	MCA	IBMFDX.OS2 5-11-94 102144 IBMFDX.NIF 5-18-94 12803 LTE.MSG LTEH.MSG IBMFDX.TXT	

<i>Table 19 (Page 14 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM FDDI Copper Extender ISA Adapter	ISA	IBMFDX.OS2 5-11-94 101744 IBMFDX.NIF 5-18-94 13226 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM FDDI Copper Extender MCA Adapter	MCA	IBMFDX.OS2 5-11-94 102144 IBMFDX.NIF 5-18-94 12803 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM FDDI Fiber Base ISA Adapter	ISA	IBMFDX.OS2 5-11-94 101744 IBMFDX.NIF 5-18-94 13226 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM FDDI Fiber Base MCA Adapter	MCA	IBMFDX.OS2 5-11-94 102144 IBMFDX.NIF 5-18-94 12803 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM FDDI Fiber Extender ISA Adapter	ISA	IBMFDX.OS2 5-11-94 101744 IBMFDX.NIF 5-18-94 13226 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM FDDI Fiber Extender MCA Adapter	MCA	IBMFDX.OS2 5-11-94 102144 IBMFDX.NIF 5-18-94 12803 LTE.MSG LTEH.MSG IBMFDX.TXT	
IBM Wireless ISA/MCA LAN Adapter	ISA MCA	IBMWLO.OS2 9-30-94 43064 IBMWLB.OS2 9-30-94 54328 IBMWLO.NIF IBMWLB.NIF	
IBM Wireless PCMCIA LAN Adapter	PCMCIA	IBMWLO.OS2 9-30-94 43064 IBMWLO.NIF	
IBM wiReless LAN ISA Adapter	ISA	IRMAC.OS2 IRMACISA.NIF IR0.MSG IR0H.MSG IRMAC.TXT	
IBM wiReless LAN MCA Adapter	MCA	IRMAC.OS2 IRMACMCA.NIF IR0.MSG IR0H.MSG IRMAC.TXT	
IBM wiReless LAN PCMCIA Adapter	PCMCIA	IRMAC.OS2 IRMACPCM.NIF IR0.MSG IR0H.MSG IRMAC.TXT	

<i>Table 19 (Page 15 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM Infrared NDIS MAC Driver for the ThinkPad 755	n/a	irdndis.os2 09-26-95 27764 irdndis.nif 09-26-95 1999 irdd.sys 11-03-95 152668 irdnds.exe 09-26-95 8313 ird.msg 09-26-95 587 irdh.msg 09-26-95 1144 irdndis.txt 09-26-95 1842	
IBM PC Network Adapter II-Frequency 2	ISA	IBMNET.OS2 IBMNET.NIF LT1.MSG LT1H.MSG	
IBM PC Network Adapter II-Frequency 3	ISA	IBMNET.OS2 IBMNET.NIF LT1.MSG LT1H.MSG	
IBM PC Network Baseband Adapter	ISA	IBMNET.OS2 IBMNET.NIF LT1.MSG LT1H.MSG	
IBM PC Network Broadband Adapter II	ISA	IBMNET.OS2 IBMNET.NIF LT1.MSG LT1H.MSG	
IBM PC Network Adapter II/A-Frequency 2	MCA	IBMNETA.OS2 IBMNETA.NIF LT1.MSG LT1H.MSG	
IBM PC Network Adapter II/A-Frequency 3	MCA	IBMNETA.OS2 IBMNETA.NIF LT1.MSG LT1H.MSG	
IBM PC Network Baseband Adapter/A	MCA	IBMNETA.OS2 IBMNETA.NIF LT1.MSG LT1H.MSG	
IBM PC Network Broadband Adapter II/A	MCA	IBMNETA.OS2 IBMNETA.NIF LT1.MSG LT1H.MSG	
IBM Advanced 3278/79 Emulation Adapter	ISA	IBMXLN.OS2 IBMXLN.NIF LT3.MSG LT3H.MSG	
IBM 3270 Connection, DFT	MCA	IBMXLN.OS2 IBMXLN.NIF LT3.MSG LT3H.MSG	
IBM Parallel Port Note: driver included with OS/2 Warp Connect	PPOINT	PRANDIS.OS2 PRANDIS.NIF PRANDISC.EXE PRN.MSG PRNH.MSG PRANDIS.TXT	

<i>Table 19 (Page 16 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
IBM Parallel Port Note: driver included with OS/2 Warp Server	PPORT	PMAC.OS2 10-30-95 28212 PMAC.NIF 10-27-95 2127 PMAC.MSG 10-26-95 821	
Intel Corporation			
Intel EtherExpress 16C (PCLA8100) Note: LAN Adapter Driver and Option Diskette for ISA and MCA Computers diskette from Intel.	ISA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress FlashC (PCLA8105)	ISA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress 16 (PCLA8110)	ISA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress Flash (PCLA8115)	ISA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress 16TP (PCLA8120)	ISA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress FlashTP (PCLA8125)	ISA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress MCA (MCLA8110)	MCA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress MCATP (MCLA8120)	MCA	EXP16.OS2 EXP16.NIF EXP16.TXT	
Intel EtherExpress PRO/10 LAN Adapter (PCLA8200A)	ISA	EPRO.OS2 05-09-95 16484 EPROEOS2.NIF 05-09-95 670	
Intel EtherExpress PRO/10 PCI	PCI	EPRO.OS2 05-09-95 16484 EPROEOS2.NIF 05-09-95 670	
Intel EtherExpress PRO/100 EISA	EISA	E100.OS2 05-10-95 21929 E100EOS2.NIF 02-10-95 1659	
Intel EtherExpress PRO/100 PCI (PILA8465)	PCI	E100.OS2 05-10-95 21929 E100EOS2.NIF 02-10-95 1659	

<i>Table 19 (Page 17 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Intel TokenExpress ISA/16S (PCLA8130A) Note: This card is auto-detected as the Olicom T/R card. The user should reject this and configure for Intel instead. Note: For all Intel TokenExpress cards, the file names used for the drivers and NIFs included with LS 4.0 differ from those used by Intel and delivered on their product diskettes. Note: 2 LAN Adapter Diskettes from Intel.	ISA	INTEL16.OS2 (OLITOK16.OS2) INTEL16.NIF (OLITOK16.NIF) INTEL.TXT	
Intel TokenExpress 16/4 LAN Adapter for EISA (EILA8235)	EISA	INTEL16.OS2 (OLITOK16.OS2) INTEL16.NIF (OLITOK16.NIF) INTEL.TXT	
Intel TokenExpress EISA/32 LAN Adapter (EILA8245)	EISA	INTEL32.OS2 (OLITOK32.OS2) INTEL32.NIF (OLITOK32.NIF) INTEL.TXT	
Kingston Technology			
Kingston Ethernet PCI (KNE40BT)	PCI	KTC40.OS2 07-21-95 16060 KTC40.NIF 08-14-95 936	
Kingston EtherRx LC ISA Combo (KNE2021LC)	ISA	KTC2000.OS2 03-28-95 7538 KTC2000.NIF 05-25-95 1317	
Kingston EtherRx LC ISA TP (KNE2000TLC)	ISA	KTC2000.OS2 03-28-95 7538 KTC2000.NIF 05-25-95 1317	
Kingston TokenRx ISA 16/4 Token-Ring Adapter	ISA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
Kingston TokenRx MC 16/4 Token-Ring Adapter	MCA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
LinkSys			
LinkSys Ether16 LAN Card Combo (LNE2000)	ISA	METH16.OS2 3-10-93 9126 METH16.NIF (not avail)	
LinkSys EtherPCI LAN Card (LNEPCI)	PCI	PCI21X4.OS2 2-28-95 13903 DC2IBM.NIF 6-2-95 581	
Madge Networks, LTD. Note: There are two drivers for the Madge Smart Ringnode family. The SMARTND.OS2 driver is intended for OS/2 1.3+. The MDGND.OS2 driver is intended for OS/2 2.0+, and is a higher performance driver.			

<i>Table 19 (Page 18 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Madge Smart 16/4 AT PLUS Ringnode (52-03)	ISA	SMARTND.OS2 5-3-94 68182 MDGND.OS2 5-18-94 76374 SMARTND.NIF MDGND.NIF	
Madge Smart 16/4 ISA Client Plus Ringnode (22-01)	ISA	SMARTND.OS2 SMARTND.NIF MDGND.NIF	
Madge Smart 16/4 ISA Client PnP Ringnode	ISA	SMARTND.OS2 SMARTND.NIF MDGND.NIF	
Madge Smart 16/4 EISA Ringnode (52-08)	EISA	SMARTND.OS2 SMARTND.NIF	
Madge Smart 16/4 MC Ringnode (54-08)	MCA	SMARTND.OS2 SMARTND.NIF	
Madge Smart 16/4 MC32 Ringnode (54-09)	MCA	SMARTND.OS2 SMARTND.NIF	
Madge Smart 16/4 PCMCIA Ringnode (20-00)	PCMCIA	SMARTND.OS2 07-17-95 82006 MDGND.OS2 3-2-94 93782 SMARTND.NIF 08-25-95 5977 SND.MSG 01-24-95 1622 MDGND.NIF 3-3-95 5122	
Madge Straight Blue 16/4 ISA (62-01)	ISA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
Madge Straight Blue ISA Plus Blue Box (62-02)	ISA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
Madge Straight Blue MC Blue Box (64-01)	MCA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
Madge Blue+ 16/4 ISA Adapter	ISA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
Madge Smart 100 AT Ringnode	ISA	MDGFND.OS2 3-28-94 89726 MDGFND.NIF MDGFNODE.BIN 3-28-94 144287	
Madge Smart 100 EISA Ringnode	EISA	MDGFND.OS2 7-30-94 54214 MDGFND.NIF MDGFNODE.BIN 7-30-94 122600	
NCR Corporation			

<i>Table 19 (Page 19 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
NCR Corporation StarLAN Token-Ring ISA	ISA	STRN.OS2 1-29-94 45114	
NCR Corporation StarLAN Token-Ring MCA	MCA	STRN.OS2 1-29-94 45114	
NCR Corporation WaveLAN Adapter		ONCRWL02.OS2 ONCRWL02.NIF	
Olicom USA, Inc.			
Olicom USA, Inc. ISA 16/4 Adapter (OC-3117) Note: for LAN Distance, card must have accelerator chip and LAN Distance must have APAR IC08555.	ISA	OLITOK16.OS2 9-8-95 63797 V7.31 OLITOK.NIF 5-2-95 6746	
Olicom USA, Inc. ISA 16/4 Adapter (OC-3118)	ISA	OLITOK16.OS2 OLITOK.NIF	
Olicom USA, Inc. EISA 16/4 Adapter (OC-3133)	EISA	OLITOK16.OS2 OLITOK.NIF	
Olicom USA, Inc. EISA/32 Adapter (OC-3135) Note: for LAN Distance, card must have accelerator chip and LAN Distance must have APAR IC08555.	EISA	OLITOK32.OS2 3-2-95 52844 v2.02 OLITOK32.NIF 7-5-94 4196	
Olicom USA, Inc. MCA 16/4 Adapter (OC-3129)	MCA	OLITOK16.OS2 OLITOK.NIF	
Olicom USA, Inc. PCI 16/4 Adapter	PCI	OLITOK16.OS2 02-14-95 60246 OLITOK.NIF 01-12-95 6746	
Olicom USA, Inc. Pocket Token-Ring Adapter (OC-3210)	PPOINT	OLITOKP.OS2 8-19-93 51334 OLITOKP.NIF 9-14-93 5433	
Olicom USA, Inc. Token-Ring PCMCIA Card (OC-3220) Note: Use the PCMCIA Card enabler from Olicom.	PCMCIA	OLITOK16.OS2 OLITOKCE.NIF OCTENABL.OS2 4-5-94 11821	
Proteon			
Proteon ProNET/E PCI Ethernet (p1670)	PCI	ETHPCI.OS2 12-28-95 10844 PRO16700.NIF 01-23-95 2574	
Proteon p1892plus ProNET - 4/16 Plus	MCA	NDIS89XR.OS2 2-16-94 15342 LS189XR.NIF 2-23-94 1162	
Proteon p1392plus ProNET - 4/16 Plus	ISA	NDIS39XR.OS2 2-16-94 15211 LS129XR.NIF 2-23-94 3743	

<i>Table 19 (Page 20 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Proteon p1393 Token-Ring	ISA	NDIS39XR.OS2 06-06-95 16590 LSI39XR.NIF 02-10-95 3744	
Proteon p1990plus ProNET - 4/16 Plus	EISA	NDIS99XR.OS2 2-16-94 15151 LS199XR.NIF 2-23-94 1125	
Racal InterLan			
Racal InterLan EtherBlaster TP-8INT (163-3184) Note: The current NIF from Racal InterLan for the EtherBlaster NICs is not compatible with LAPS. The user must create a NIF manually. See the detailed instructions in READMAC.TXT.	ISA	NI6510.OS2 9-17-93 41363 NI6510.NIF	
Racal InterLan AT-TP (163-3118) Note: The current NIF from Racal InterLan AT NICs is not compatible with LAPS. The user must create a NIF manually. See the detailed instructions in READMAC.TXT.	ISA	ILANAT.OS2 12-23-92 9916 ILANAT.NIF	
Racal InterLan NI5210-16 (163-0610) Note: The current NIF from Racal InterLan for NI5210 NICs is not compatible with LAPS. The user must create a NIF manually. See the detailed instructions in READMAC.TXT.	ISA	NI5210.OS2 6-3-92 9216 NI5210.NIF	
Racal InterLan ES3210 Note: The current NIF from Racal InterLan for ES3210 NICs is not compatible with LAPS. The user must create a NIF manually. See the detailed instructions in READMAC.TXT.	EISA	ES3210.OS2 10-12-93 11682 ES3210.NIF	
Racal InterLan ES3210-TP (163-3160)	EISA	ES3210.OS2 ES3210.NIF	
Racal InterLan MCA (163-3142) Note: The current NIF from Racal InterLan for MCA (9210) NICs is not compatible with LAPS. The user must create a NIF manually. See the detailed instructions in READMAC.TXT. Note: Some problems on the driver dated 5-18-93 7804 bytes. Use the driver dated 6-3-92.	MCA	NI9210.OS2 6-3-92 9214 NI9210.NIF	
Racal InterLan MCA-TP (163-3143) Note: See notes for Racal InterLan MCA, above.	MCA	NI9210.OS2 6-3-92 9214 NI9210.NIF	
Racal InterLan PCI T2 (163-3215)	PCI	ILANPCI.OS2 03-30-94 18247	

<i>Table 19 (Page 21 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Racal InterLan T/R 16/4 ISA (167-3193)	ISA	RIC16NDS.OS2 4-22-94 8256 RIC16NDS.NIF 4-22-94 2434	
Racal InterLan T/R 16/4 MCA (163-3137)	MCA	RDC16NDS.OS2 6-29-93 8756 RDC16NDS.NIF 7-1-93 2434	
Racore Computer Products, Inc.			
Racore Computer Products, Inc. Token-Ring ISA (M8119)	ISA	RTR16NDS.OS2 RTR16NDS.NIF RTR16NDS.MSG RTR16NDS.TXT	
Racore Computer Products, Inc. Token-Ring MC	MCA	RTR16NDS.OS2 RTR16NDS.NIF RTR16NDS.MSG RTR16NDS.TXT	
Standard Microsystems Corporation			
Note: Includes some adapter cards previously sold through Western-Digital			
SMC EtherCard PLUS (8003EB)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard PLUS/A (8003E/A)	MCA	SMC8000.OS2 ETHOS2MC.NIF SMC8000.TXT	
SMC EtherCard PLUS Elite (8003EP) Note: The file MACH.MSG might not appear on the SMC driver diskette V. 4.6. The user needs to move the driver files manually to \ibmcom\macs, or modify the NIF to not reference MACH.MSG then install. Note: The PLUS Elite family may also work on the MACWD.OS2 which is delivered with LAPS. However, the SMC MAC driver is preferred.	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard PLUS Elite 10T (8003WC)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard PLUS Elite 16T (8013WC)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard PLUS Elite 16 (8013EPC)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard PLUS Elite 16 Combo (8013EWC)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	

<i>Table 19 (Page 22 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
SMC EtherCard PLUS Elite/A (8013EP/A)	MCA	SMC8000.OS2 ETHOS2MC.NIF SMC8000.TXT	
SMC EtherCard PLUS Elite 10 T/A (8013WP/A)	MCA	SMC8000.OS2 ETHOS2MC.NIF SMC8000.TXT	
SMC EtherCard Elite 16 Ultra (8216)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard Elite 16C Ultra (8216C)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard Elite 16T Ultra (8216T)	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherEZ 10BASE-T ISA (8416T) Note: Plug and Play must be disabled.	ISA	SMC8000.OS2 ETHOS2AT.NIF SMC8000.TXT	
SMC EtherCard Elite32C Ultra (82M32C) Note: To operate with LAN Distance, one must disable the busmaster mode. Otherwise, the system will not boot correctly.	EISA busmaster	SMC8232.OS2 SMCOS2E.NIF SMC8232.TXT	
SMC EtherPower 10BASE-T PCI Ethernet Adapter (8432T)	PCI		
SMC EtherPower Combo PCI (8432BT) Note: for BNC, add the following to the protocol.ini: SIA_Mode = BNC	PCI busmaster	SMCPWR.OS2 1-5-95 13355 SMC93320.NIF 2-2-95 198	
SMC EtherPower 10/100 Fast Ethernet PCI (9332DST)	PCI busmaster	SMCPWR.OS2 1-5-95 13355 SMC93320.NIF 2-2-95 198	
SMC 9000	n/a	SMC9X.OS2 09-29-95 16964 SMC9000.NIF 09-29-95 1239	
SMC TokenCard Elite (8115T)	ISA	SMC8100.OS2 TOKOS2AT.NIF SMC8100.TXT	
SMC TokenCard Elite/A (8115T/A)	MCA	SMC8100.OS2 TOKOS2MC.NIF SMC8100.TXT	
SMC TokenCard Elite Master32 (83M32)	EISA	SMC8332.OS2 SMCOS2ET.NIF SMC8332.TXT	
TDK Corporation			

<i>Table 19 (Page 23 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
TDK LAN LAC-CD021 PCMCIA Ethernet Adapter	PCMCIA	TDKCD02.OS2 5-17-95 14684 TDKCD02.NIF 5-17-95 1607	
Texas Instruments, Inc.			
Texas Instruments, Inc. TokenLite Token-Ring Adapter	ISA	TR2KNDIS.OS2 TR2KNDIS.NIF TR2KNDIS.MSG TR2KNDIS.TXT	
Thomas-Conrad			
Thomas-Conrad Ethernet PCI (TC5048-T2)	PCI	TCE32PCW.OS2 05-15-95 13903 TCE32PCW.NIF 06-13-95 1798	
Thomas-Conrad 16/4 Token-Ring Adapter/AT (TC4045) Note: The file EAGLEMAC.BIN must be in the ROOT directory of the boot drive. This file is found in the \IBMCOM\MACS directory by default, and must be copied to the root.	ISA	TCCTOK.OS2 TCCTOK.NIF EAGLEMAC.BIN	
Thomas-Conrad 16/4 Token-Ring Adapter/MC (TC4046) Note: The file EAGLEMAC.BIN must be in the ROOT directory of the boot drive. This file is found in the \IBMCOM\MACS directory by default, and must be copied to the root.	MCA	TCCTOK.OS2 TCCTOK.NIF EAGLEMAC.BIN	
Thomas-Conrad Tropic 16/4 Token-Ring Adapter/AT (TC4043)	ISA	IBMTOK.OS2 IBMTOKC.NIF LT2.MSG LT2H.MSG	
Ungermann-Bass			
Ungermann-Bass NIUpc Adapter	ISA	UBNEI.OS2 UBNEIPC.NIF IBI.MSG UBIH.MSG	
Ungermann-Bass NIUps Adapter	MCA	UBNEI.OS2 UBNEIPS.NIF IBI.MSG UBIH.MSG	
Xircom			
Xircom External Token-Ring Adapter (ET16BU)	PPORT	SMARTND.OS2 9-8-92 88150 XIRTOK.NIF 8-17-92 901 TRSETUP.OS2 9-8-92 37742	
Xircom Pocket Token Ring Adapter III (PT3-16CTP)	PPORT	SMARTND.OS2 10-7-93 72278 XIRTOK.NIF 11-1-93 2402	

<i>Table 19 (Page 24 of 24). Network Interface Card MAC Drivers for OS/2 NDIS 2.0.1</i>			
Vendor / Product Name	Bus Type	OS/2 MAC Driver	DOS MAC Driver
Xircom CreditCard Token Ring Adapter (CT-16CTP)	PCMCIA	CTND.OS2 12-2-93 68694 CTOS2V2.NIF 12-9-93 848	
Xircom Pocket Ethernet Adapter III (PE3-10BT)	PPORT	PE3NDIS.OS2 10-4-93 28736 PE3OS2V2.NIF 11-9-93 194	
Xircom Pocket Ethernet Adapter III (PE3-10BC)	PPORT	PE3NDIS.OS2 10-4-93 28736 PE3OS2V2.NIF 11-9-93 194	
Xircom Pocket Ethernet Adapter III (PE3-10BX)	PPORT	PE3NDIS.OS2 2-12-93 22864 PE3OS2V2.NIF 2-16-93 197	
Xircom CreditCard Ethernet Adapter (CE-10BC) Note: Disable IBM Card and Socket Services. (REM from config.sys two (or more) lines: PCMCIA.SYS and IBM2SS01.SYS).	PCMCIA	CENDIS.OS2 4-24-94 12800 CEOS2V2.NIF 4-22-94 847	
Xircom CreditCard Ethernet Adapter (CE-10BT/A)	PCMCIA	CE2NDIS.OS2 10-05-94 18978 CE2OS2.NIF 11-03-94 706	
Xircom PS-CreditCard Ethernet Adapter (PS-CE2-10BT) Note: Disable IBM Card and Socket Services. (REM from config.sys two (or more) lines: PCMCIA.SYS and IBM2SS01.SYS).	PCMCIA	CE2NDIS.OS2 10-05-94 18978 CE2OS2.NIF 11-03-94 706	

All trademarks contained herein are the property of their respective trademark owners.

E.2 Network Interface Card Support Matrix for OS/2 Warp V3 LAN Systems

The following table does not imply testing or certification by IBM. Where appropriate, certification of these adapters is indicated in this table. LAN Systems support is indicated as follows:

- Yes - means driver is included in the LAN Systems product, and the product is supported using the adapter card. The adapter card and the driver are supported by the vendor.
- Yes-DD - means the LAN Systems product is supported using the adapter card, but the driver is not included and must be obtained either from the OS/2 Supplemental NIC Driver Diskettes or from the vendor of the

adapter card. To locate a copy of the supplemental diskettes for OS/2 or DOS, please refer to the instructions included under 'Resource Information for NIC Support.'

- Vendor - means the LAN Systems product is supported using the adapter card, but the driver is not included and must be obtained from the vendor of the adapter card.
- **ITL** - means IBM tested the card with the LS product in the Test and Approved for IBM LAN Systems program.
- The NICs identified under Warp Connect are supported with the LAN Requester 4.0, TCP/IP 3.0, and OS/2 Peer components.
- The NICs identified under Warp Server are supported with the LAN Server, OS/2 LAN Requester, and TCP/IP components.

<i>Table 20 (Page 1 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
3Com Corporation				
3Com EtherLink II (3C503)	Yes	Yes	Yes	
3Com EtherLink II-16 (3C503-16)	Yes	Yes	Yes	
3Com EtherLink II-16-TP (3C503-16-TP)	Yes	Yes	Yes	
3Com EtherLink 16 (3C507)	Vendor	Vendor		
3Com EtherLink 16-TP (3C507-TP)	Vendor	Vendor		
3Com EtherLink/MC (3C523B)	Yes	Yes	Yes	
3Com EtherLink/MC-TP (3C523B-TP)	Yes	Yes	Yes	
3Com EtherLink MC-32 (3C527B)	Vendor	Vendor	Vendor	
3Com EtherLink III (3C509)	Yes	Yes		Yes
3Com EtherLink III (3C509B)	Yes	Yes	Vendor	Yes
3Com EtherLink III-Combo (3C509-COMBO)	Yes	Yes	Yes	
3Com EtherLink III-Combo (3C509B-COMBO)	Yes	Yes	Vendor	Yes
3Com EtherLink III-TP (3C509-TP)	Yes	Yes	Yes	Yes
3Com EtherLink III-TP (3C509B-TP)	Yes	Yes	Vendor	Yes
3Com EtherLink III-TPO (3C509-TPO)	Yes	Yes	Yes	
3Com EtherLink III-TPO (3C509B-TPO)	Yes	Yes	Vendor	Yes
3Com EtherLink III-EISA (3C579)	Yes	Yes	Vendor	Yes
3Com EtherLink III-EISA-TP (3C579-TP)	Yes	Yes		Vendor
3Com EtherLink III-MCA (3C529)	Yes	Yes		Vendor
3Com EtherLink III-MCA-TP (3C529-TP)	Yes	Yes	Vendor	
3Com EtherLink III-PCMCIA-TP (3C589-TP)	Vendor	Vendor*		
3Com EtherLink III-PCMCIA-Combo (3C589B-Combo)	Vendor ITL	Vendor*	Vendor	

<i>Table 20 (Page 2 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
3Com EtherLink III LAN PC Card (3C589C)	Vendor ITL	Vendor*	Vendor	
3Com EtherLink III LAN + Modem PC Card (3C562) (LAN ONLY)	Vendor ITL	Vendor*	Vendor	
3Com EtherLink III PCI 10BASE-T Network Adapter (3C590-TPO)		Vendor		
3Com Fast EtherLink PCI 10/100 BASE-T Network Adapter (3C595-TX)		Vendor		
3Com Fast EtherLink EISA 10/100 BASE-T Network Adapter (3C597-TX)		Vendor		
3Com TokenLink III (3C619)	Yes	Yes		Yes
3Com TokenLink III EISA (3C679)	Yes	Yes		
3Com TokenLink III MCA (3C629)	Yes	Yes		Yes
3Com TokenLink III 16/4 PC Card (3C689)	Vendor	Vendor*	Vendor	
Accton				
Accton EtherCombo-16 (EN1650)	Vendor	Vendor		Vendor
Accton EtherPair-16 (EN1651)	Vendor	Vendor		Vendor
Accton EtherCoax-16 (EN1652)	Vendor	Vendor		
Accton EtherCombo-32 (EN1200)	Vendor	Vendor		
Advanced Micro Devices, Inc.				
AMD PCnet-32 Ethernet Adapter	Vendor	Vendor	Vendor	Vendor
AMD PCnet-ISA II Ethernet Adapter	Vendor	Vendor	Vendor	Vendor
AMD PCnet-PCI Ethernet Adapter	Vendor	Vendor	Vendor	Vendor
Allied Telesis				
Allied Telesis Ethernet Adapter Card ISA (AT1500-Plus)	Vendor	Vendor		
Allied Telesis AT1700 Plus ISA	Vendor	Vendor		
Allied Telesis AT1720 Plus MCA	Vendor	Vendor		
Artisoft				
Artisoft NodeRunner/SI 2000/C	Yes	Yes		
Artisoft NodeRunner/SI 2000/T	Yes	Yes	Vendor	
Artisoft NodeRunner/SI 2000/A	Yes	Yes	Vendor	
Artisoft NodeRunner/SI 2000M/TC	Yes	Yes	Vendor	
Artisoft LANTastic NodeRunner 2000/C	Yes	Yes	Vendor	
Artisoft LANTastic NodeRunner 2000/T	Yes	Yes		
Artisoft LANTastic NodeRunner 2000/A	Yes	Yes	Vendor	
Artisoft LANTastic NodeRunner 2000M/TC	Yes	Yes	Vendor	
Asante				
Asante EtherPaC 2000+3	Vendor	Vendor		

<i>Table 20 (Page 3 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
Asante EtherPaC 2000+N	Vendor	Vendor		
Asante EtherPaC 2000+T	Vendor	Vendor		
Cabletron Corporation				
Cabletron Ethernet DNI Adapter (E1112)	Yes	Yes		
Cabletron Ethernet DNI Adapter (E1119)	Yes	Yes		
Cabletron Ethernet DNI Adapter (E2112)	Yes	Yes	Vendor	
Cabletron Ethernet DNI Adapter (E2119)	Yes	Yes	Vendor	
Cabletron Ethernet DNI Adapter (E3112)	Yes	Yes	Vendor	
Cabletron Ethernet DNI Adapter (E3119)	Yes	Yes	Vendor	
Cabletron Token-Ring DNI Adapter (T2015)	Yes	Yes	Vendor	
Cabletron Token-Ring DNI Adapter (T3015)	Yes	Yes		
CeLAN				
CeLAN FlexLINK - EPCiplus (9910EPCI-B)	Vendor	Vendor		
Cogent Data Technologies, Inc.				
Cogent eMASTER+ EM960 PCI Ethernet Adapter (EM960C)	Vendor	Vendor		
Cogent EM100 PCI FAST Ethernet Adapter		Vendor		
Compaq				
Compaq NetFlex-2 ENET-TR Controller	Vendor	Vendor		
Cray Communications				
Cray Communications ScaNet Network Interface Adapter-ISA	Yes	Yes		
Cray Communications ScaNet Network Interface Adapter-MCA	Yes	Yes		
Digital Communications Associates				
DCA ClassicBlue MC 4/16 Token-Ring Adapter	Yes	Yes		
DCA IRMAtrac EISA	Vendor	Vendor	Vendor	
DCA IRMAtrac Token-Ring Adapter/Convertible-MCA	Vendor	Vendor		
Digital Equipment Corporation				
Digital EtherWorks 3 Turbo TP (DE204-AA)			Vendor	
Digital EtherWorks Turbo 435 PCI	Vendor	Vendor		
Digital Semiconductor				
Digital EB40-DECchip 21040 Evaluation Board	Vendor	Vendor	Vendor	Vendor
Digital EB140-DECchip 21140 Evaluation Board	Vendor	Vendor	Vendor	Vendor
D-Link				
D-Link Ethernet Interface Card for the PC XT/AT (DE-220C)	Vendor	Vendor		

<i>Table 20 (Page 4 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
D-Link Ethernet Interface Card for the PC XT/AT (DE-220CAT)	Vendor	Vendor		
D-Link Ethernet Interface Card for the PC XT/AT (DE-220CT)	Vendor	Vendor	Vendor	
D-Link Ethernet Interface Card for the PC XT/AT (DE-220T)	Vendor	Vendor	Vendor	
D-Link Ethernet Card for EISA bus PC (DE-400)	Vendor	Vendor	Vendor	
D-Link Ethernet VESA Combo Card (DE-500CAT)	Vendor	Vendor		
D-Link Ethernet PCI (DE-530CT)	Vendor	Vendor		
D-Link Token-Ring Adapter for the PC/AT and PS/2 (DT-220)			Vendor	
Eagle Technology				
Eagle Novell NE2000	Yes	Yes		
Eagle Novell NE2000T	Yes	Yes	Yes	
Eagle Novell NE2000plus	Yes	Yes	Yes	
Eagle Novell NE2000Tplus	Yes	Yes	Yes	Vendor
Eagle Novell NE2000plus-3	Yes	Yes	Yes	Vendor
Eagle EtherXpert EP2000plus	Yes	Yes	Yes	
Eagle EtherXpert EP2000Tplus	Yes	Yes	Yes	
Eagle Novell NE3210	Yes	Yes	Vendor	Vendor
Eagle EtherXpert EP3210	Yes	Yes	Vendor	
Eagle Novell NE/2T			Yes	
Hewlett-Packard				
Hewlett-Packard 27247B	Vendor	Vendor	Vendor	
Hewlett-Packard PCLAN Adapter/16 PLUS (27252A)	Vendor	Vendor		
Hewlett-Packard JP2405A			Vendor	
Hewlett-Packard 10/100VG PC LAN ISA Adapter (J2573A)	Vendor	Vendor		
Hewlett-Packard 10/100VG PC LAN EISA Adapter (J2577A)	Vendor	Vendor		
Hewlett-Packard 10/100VG PC LAN PCI Adapter (J2585A)	Vendor	Vendor		
IBM Corporation				
IBM LAN Adapter for Ethernet (48G7169)	Yes	Yes	Yes	Yes
IBM LAN Adapter for Ethernet CX (60G615)	Yes	Yes		Yes
IBM LAN Adapter for Ethernet TP (60G0605)	Yes	Yes		Yes
IBM EtherJet ISA Adapter	Vendor ITL	Vendor	Vendor	Vendor

<i>Table 20 (Page 5 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
IBM EtherJet 10-BASE-T ISA Adapter	Vendor ITL	Vendor	Vendor	Vendor
IBM Ethernet Network Adapter 10BaseT 66G0939 (JAPAN ONLY)		Yes	Vendor	
IBM Ethernet Network Adapter 10Base2 66G0943 (JAPAN ONLY)		Yes	Vendor	
IBM Credit Card Adapter for Ethernet 10B2 (0933280)	Yes	Yes*		
IBM Credit Card Adapter for Ethernet 10BT (0933290)	Yes	Yes*		
IBM Credit Card Adapter II for Ethernet 10B2 (0934330)	Yes	Yes*	Vendor	
IBM Credit Card Adapter II for Ethernet 10BT (0934331)	Yes	Yes*	Vendor	
IBM Adapter/A for Ethernet Networks (6451091)	Yes	Yes	Vendor	Yes
IBM Adapter/A for Ethernet Twisted-Pair Networks (6451136)	Yes	Yes	Vendor	
IBM Ethernet Network Adapter/A 10Base2/5 35G2793 (JAPAN ONLY)		Yes	Yes	
IBM Ethernet Network Adapter/A 10Base5/T 35G2806 (JAPAN ONLY)		Yes	Yes	
IBM LAN Adapter/A for Ethernet (48G7171)	Yes	Yes	Vendor	Yes
IBM EtherStreamer MC 32 Adapter (59G9066)	Yes	Yes		Vendor
IBM EtherStreamer MC 32 Adapter (74G0883) (JAPAN ONLY)		Yes	Vendor	
IBM Dual EtherStreamer MC 32 Adapter (73G7136)	Yes	Yes		
IBM Ethernet Quad-BT PeerMaster Server Adapter (06H5184)	Vendor	Vendor		
IBM Ethernet Quad-B2 PeerMaster Server Adapters (06H6041)	Vendor	Vendor		
IBM Token-Ring Network PC Adapter	Yes	Yes		
IBM Token-Ring Network PC Adapter II	Yes	Yes	Yes	
IBM Token-Ring Network 16/4 Adapter	Yes	Yes	Yes	Yes
IBM Token-Ring Network 16/4 ISA-16 Adapter (73G2032)	Yes	Yes	Yes	
IBM Token-Ring Network 16/4 Adapter II	Yes	Yes		
IBM Auto 16/4 Token-Ring ISA Adapter (92G7632)	Yes	Yes	Vendor	Yes
IBM 16/4 Busmaster EISA Adapter (1051712)	Yes	Yes		
IBM Token-Ring 16/4 Credit Card Adapter (0933462)	Yes	Yes*		

<i>Table 20 (Page 6 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
IBM Token-Ring 16/4 Credit Card Adapter II (0933931)	Yes	Yes*	Vendor	
IBM PCMCIA Token-Ring Adapter (04H6922)	Yes	Yes*		
IBM Token-Ring Network Adapter/A (69X8138)	Yes	Yes	Yes	Yes
IBM Token-Ring Network 16/4 Adapter/A (16F1133)	Yes	Yes	Yes	
IBM Token-Ring Network 16/4 Adapter/A (74F9410)	Yes	Yes	Yes	Yes
IBM Auto 16/4 Token-Ring MC Adapter (92G7682)	Yes	Yes	Vendor	Yes
IBM Token-Ring Network 16/4 Busmaster Server Adapter/A (74F4140)	Yes	Yes		
IBM LANStream MC 16 Adapter (74G0801)	Yes	Yes		
IBM LANStream MC 32 Adapter (74G0103)	Yes	Yes		
IBM Auto LANStream MC 32 Adapter (60G1592)	Yes	Yes	Vendor	Vendor
IBM Dual LANStream MC 32 Adapter (73G7137)	Yes	Yes		Vendor
IBM Auto LANStream PCI Adapter (04H8095)	Vendor	Yes	Vendor	Yes
IBM FDDI Copper Base ISA Adapter	Yes	Yes	Vendor	
IBM FDDI Copper Extender ISA Adapter	Yes	Yes	Vendor	
IBM FDDI Fiber Base ISA Adapter	Yes	Yes	Vendor	
IBM FDDI Fiber Extender ISA Adapter	Yes	Yes	Vendor	
IBM FDDI Copper Base MCA Adapter	Vendor	Vendor	Vendor	
IBM FDDI Copper Extender MCA Adapter	Vendor	Vendor	Vendor	
IBM FDDI Fiber Base MCA Adapter	Vendor	Vendor	Vendor	
IBM FDDI Fiber Extender MCA Adapter	Vendor	Vendor	Vendor	
IBM Wireless ISA/MCA LAN Adapter	Yes	Yes	Vendor	
IBM Wireless PCMCIA LAN Adapter	Yes	Yes*	Vendor	
IBM wIReless LAN ISA Adapter	Yes	Yes		
IBM wIReless LAN MCA Adapter	Yes	Yes		
IBM wIReless LAN PCMCIA Adapter	Yes	Yes*		
IBM Infrared NDIS MAC Driver for the ThinkPad 755	Vendor	Yes*		
IBM PC Network Adapter II-Frequency 2	Yes			
IBM PC Network Adapter II-Frequency 3	Yes			
IBM PC Network Baseband Adapter	Yes			
IBM PC Network Broadband Adapter II	Yes			
IBM PC Network Adapter II/A-Frequency 2	Yes			

<i>Table 20 (Page 7 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
IBM PC Network Adapter II/A-Frequency 3	Yes			
IBM PC Network Baseband Adapter/A	Yes			
IBM PC Network Broadband Adapter II/A	Yes			
IBM Advanced 3278/79 Emulation Adapter	Yes			
IBM 3270 Connection, DFT	Yes			
IBM Parallel Port	Yes	Yes*		
Intel Corporation				
Intel EtherExpress 16C (PCLA8100)	Yes	Yes	Yes	
Intel EtherExpress FlashC (PCLA8105)	Yes	Yes	Yes	
Intel EtherExpress 16 (PCLA8110)	Yes	Yes	Yes	
Intel EtherExpress Flash (PCLA8115)	Yes	Yes		
Intel EtherExpress 16TP (PCLA8120)	Yes	Yes	Yes	
Intel EtherExpress FlashTP (PCLA8125)	Yes	Yes	Yes	
Intel EtherExpress MCA (MCLA8110)	Yes	Yes	Yes	
Intel EtherExpress MCATP (MCLA8120)	Yes	Yes	Yes	
Intel EtherExpress PRO/10 LAN Adapter (PCLA8200A)	Vendor	Vendor		
Intel EtherExpress PRO/10 PCI		Vendor		
Intel EtherExpress PRO/100 EISA		Vendor		
Intel EtherExpress PRO/100 PCI (PILA8465)		Vendor		
Intel TokenExpress ISA/16S (PCLA8130A)	Yes	Yes	Yes	
Intel TokenExpress 16/4 LAN Adapter for EISA (EILA8235)	Yes	Yes	Yes	
Intel TokenExpress EISA/32 LAN Adapter (EILA8245)	Yes	Yes	Vendor	
Kingston Technology				
Kingston EtherRx PCI Ethernet Adapter (KNE40BT)	Vendor	Vendor	Vendor	Vendor
Kingston EtherRx LC ISA Combo (KNE2021LC)	Vendor	Vendor	Vendor	Vendor
Kingston EtherRx LC ISA TP (KNE2000TLC)	Vendor	Vendor		
Kingston Ethernet PC Card	Vendor ITL	Vendor*	Vendor	
Kingston TokenRx ISA 16/4 Token-Ring Adapter	Vendor	Vendor	Vendor	
Kingston TokenRx MC 16/4 Token-Ring Adapter	Vendor	Vendor	Vendor	
Kingston TokenRx PCMCIA 16/4 Adapter (KTR-PCM16/4)	Vendor	Vendor*	Vendor	
LinkSys				

<i>Table 20 (Page 8 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
LinkSys Ether16 LAN Card Combo (LNE2000)	Vendor	Vendor		
LinkSys EtherPCI LAN Card (LNEPCI)	Vendor	Vendor		
Madge Networks LTD.				
Madge Smart 16/4 AT PLUS Ringnode (52-03)	Yes	Yes	Vendor	
Madge Smart 16/4 ISA Client Plus (22-01)			Vendor	
Madge Smart 16/4 ISA Client PnP Ringnode	Yes	Yes	Vendor	
Madge Smart 16/4 EISA Ringnode (52-08)	Yes	Yes	Vendor	
Madge Smart 16/4 MC Ringnode (54-08)	Yes	Yes	Vendor	
Madge Smart 16/4 MC32 Ringnode (54-09)	Yes	Yes	Vendor	
Madge Smart 16/4 PCMCIA Ringnode (20-00)	Vendor	Vendor*	Vendor	
Madge Smart 16/4 PCI Ringnode	Vendor	Vendor	Vendor	
Madge Straight Blue 16/4 ISA (62-01)	Yes	Yes		
Madge Straight Blue ISA Plus Blue Box (62-02)	Yes	Yes		
Madge Straight Blue MC Blue Box (64-01)	Yes	Yes		Yes
Madge Blue+ 16/4 ISA Adapter	Vendor	Vendor	Vendor	
Madge Smart 100 EISA Ringnode	Yes	Yes	Vendor	
Madge Smart 100 AT Ringnode	Yes	Yes	Vendor	
NCR Corporation				
NCR StarLAN Token-Ring ISA	Vendor	Vendor		
NCR StarLAN Token-Ring MCA	Vendor	Vendor		
NCR Corporation WaveLAN Adapter	Vendor	Vendor		
Olicom USA, Inc.				
Olicom USA, Inc. ISA 16/4 Adapter (OC-3117)	Yes	Yes	Yes	
Olicom USA, Inc. ISA 16/4 Adapter (OC-3118)	Vendor	Vendor	Vendor	
Olicom USA, Inc. EISA 16/4 Adapter (OC-3133)	Vendor	Vendor	Yes	
Olicom USA, Inc. EISA/32 Adapter (OC-3135)	Vendor	Vendor	Vendor	Vendor
Olicom USA, Inc. MCA 16/4 Adapter (OC-3129)	Vendor	Vendor	Yes	Vendor
Olicom USA, Inc. PCI 16/4 Adapter		Vendor		
Olicom USA, Inc. Pocket Token-Ring Adapter (OC-3210)	Vendor	Vendor*	Vendor	
Olicom USA, Inc. Token-Ring PCMCIA Card (OC-3220)	Vendor	Vendor*	Vendor	
Proteon				
Proteon ProNET/E PCI Ethernet (p1670)	Vendor	Vendor		
Proteon p1892plus ProNET - 4/16 Plus	Vendor	Vendor		
Proteon p1392plus ProNET - 4/16 Plus	Vendor	Vendor	Yes	
Proteon p1393 TokenRing ISA		Vendor		

<i>Table 20 (Page 9 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
Proteon p1990plus ProNET - 4/16 Plus	Vendor	Vendor	Vendor	
Racal InterLan				
Racal InterLan EtherBlaster TP-8INT (163-3184)	Vendor	Vendor		
Racal InterLan AT-TP (163-3118)	Vendor	Vendor		
Racal InterLan NI5210-16 (163-0610)	Vendor	Vendor		
Racal InterLan ES3210	Vendor	Vendor		
Racal InterLan ES3210-TP (163-3160)	Vendor	Vendor		
Racal InterLan MCA (163-3142)	Vendor	Vendor		
Racal InterLan MCA-TP (163-3143)	Vendor	Vendor	Vendor	
Racal InterLan PCI T2 (163-3215)	Vendor	Vendor		
Racal InterLan T/R 16/4 ISA (167-3193)	Vendor	Vendor	Vendor	
Racal InterLan T/R 16/4 MCA (163-3137)	Vendor	Vendor	Vendor	
Racore Computer Products, Inc.				
Racore Computer Products, Inc. Token-Ring ISA (M8119)	Yes	Yes		
Racore Computer Products, Inc. Token-Ring MC	Yes	Yes		
Standard Microsystems Corporation				
SMC EtherCard PLUS (8003EB)	Yes	Yes		
SMC EtherCard PLUS/A (8003E/A)	Yes	Yes		
SMC EtherCard PLUS Elite (8003EP)	Vendor	Yes		
SMC EtherCard PLUS Elite 10T (8003WC)	Vendor	Yes	Yes	Vendor
SMC EtherCard PLUS Elite 16T (8013WC)	Vendor	Yes	Yes	Vendor
SMC EtherCard PLUS Elite 16 (8013EPC)	Vendor	Yes		Vendor
SMC EtherCard PLUS Elite 16 Combo (8013EWC)	Vendor	Yes	Yes	Vendor
SMC EtherCard PLUS Elite/A (8013EP/A)	Vendor	Yes		
SMC EtherCard PLUS Elite 10 T/A (8013WP/A)	Vendor	Yes	Yes	
SMC EtherCard Elite 16 Ultra (8216)	Vendor	Yes		Vendor
SMC EtherCard Elite 16C Ultra (8216C)	Vendor	Yes		Vendor
SMC EtherCard Elite 16T Ultra (8216T)	Vendor	Yes	Vendor	Vendor
SMC EtherEZ 10BASE-T ISA (8416T)	Vendor	Yes		
SMC EtherCard Elite32C Ultra (82M32C)	Vendor	Yes		Vendor
SMC EtherPower 10BASE-T PCI Ethernet Adapter (8432T)	Vendor	Vendor		
SMC EtherPower Combo PCI Ethernet Adapter (8432BT)	Vendor	Vendor		
SMC EtherPower 10/100 Fast Ethernet PCI Adapter (9332DST)	Vendor	Vendor		

<i>Table 20 (Page 10 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
SMC 9000	<i>Vendor</i>	<i>Vendor</i>		
SMC TokenCard Elite (8115T)	Vendor	Yes	<i>Vendor</i>	
SMC TokenCard Elite/A (8115T/A)	Vendor	Yes	Vendor	
SMC TokenCard Elite Master32 (83M32)	Vendor	Yes	Vendor	
SysKonnnect, Inc.				
SK-NET FDDI-ISA Network Interface card (SK-5141)	<i>Vendor</i>	<i>Vendor</i>	<i>Vendor</i>	
SK-NET FDDI-EISA Network Interface card (SK-5341)	<i>Vendor</i>	<i>Vendor</i>	<i>Vendor</i>	
SK-NET FDDI-MCA Network Interface card (SK-5241)	<i>Vendor</i>	<i>Vendor</i>	<i>Vendor</i>	
TDK Corporation				
TDK Corporation TDKLAN LAC-CD021 PCMCIA Ethernet Adapter	<i>Vendor ITL</i>	<i>Vendor*</i>		
Texas Instruments, Inc.				
Texas Instruments, Inc. TokenLite Token-Ring Adapter	Yes	Yes		
Thomas-Conrad				
Thomas-Conrad Ethernet PCI (TC5048-T2)	Vendor	Vendor		
Thomas-Conrad 16/4 Token-Ring Adapter/AT (TC4045)	Yes	Yes	<i>Vendor</i>	
Thomas-Conrad 16/4 Token-Ring Adapter/MC (TC4046)	Yes	Yes	Vendor	
Thomas-Conrad Tropic 16/4 Token-Ring Adapter/AT (TC4043)	Yes	Yes	<i>Vendor</i>	
Ungermann-Bass				
Ungermann-Bass NIUpc Adapter	Yes	Yes		
Ungermann-Bass NIUps Adapter	Yes	Yes		
Xircom				
Xircom External Token-Ring Adapter (ET16BU)	Vendor	Vendor*		
Xircom Pocket Token Ring Adapter III (PT3-16CTP)	Vendor	Vendor*	Vendor	
Xircom CreditCard Token Ring Adapter (CT-16CTP)	Vendor	Vendor*		
Xircom Pocket Ethernet Adapter III (PE3-10BT)	Vendor	Vendor*		
Xircom Pocket Ethernet Adapter III (PE3-10BC)	Vendor	Vendor*		
Xircom Pocket Ethernet Adapter III (PE3-10BX)	Vendor	Vendor*		
Xircom CreditCard Ethernet Adapter (CE-10BC)	Vendor	Vendor*		
Xircom CreditCard Ethernet Adapter (CE-10BT/A)	Vendor	Vendor*		

<i>Table 20 (Page 11 of 11). Network Interface Card Support for OS/2 Warp LAN Systems</i>				
Vendor / Product Name	Warp Connect 3.0	Warp Server 4.0	DLS 5.0 8200	LD (WS)
Xircom PS-CreditCard Ethernet Adapter (PS-CE2-10BT)	Vendor	Vendor*		

All trademarks contained herein are the property of their respective trademark owners.

E.3 LS Product Matrix

For details regarding support on OS/2 Warp Connect and OS/2 Warp Server, please see the Warp Product Matrix.

The following table does not imply testing or certification by either IBM and/or National Software Testing Labs (NSTL). Where appropriate, certification of these adapters is indicated in this table. LAN Systems support is indicated as follows:

- Yes - means driver is included in the LAN Systems product, and the product is supported using the adapter card. The adapter card and the driver are supported by the vendor.
- Yes-DD - means the LAN Systems product is supported using the adapter card, but the driver is not included and must be obtained either from the OS/2 Supplemental NIC Driver Diskettes or from the vendor of the adapter card. To locate a copy of the supplemental diskettes for OS/2 or DOS, please refer to the instructions included under 'Resource Information for NIC Support.'
- Vendor - means the LAN Systems product is supported using the adapter card, but the driver is not included and must be obtained from the vendor of the adapter card.
- **NSTL** - means NSTL tested the card with the LS product in the NDIS Driver Compatibility Program.
- **ITL** - means IBM tested the card with the LS product in the Test and Approved for IBM LAN Systems program.

* This adapter card is supported on clients only.

<i>Table 21 (Page 1 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
3Com Corporation						
3Com EtherLink II (3C503)	Yes		Yes		Yes	
3Com EtherLink II-16 (3C503-16)	Yes	Yes	Yes	Yes	Yes	
3Com EtherLink II-16-TP (3C503-16-TP)			Yes		Yes	
3Com EtherLink 16 (3C507)			Vendor			
3Com EtherLink 16-TP (3C507-TP)			Vendor			
3Com EtherLink/MC (3C523B)	Yes		Yes		Yes	
3Com EtherLink/MC-TP (3C523B-TP)			Yes		Yes	
3Com EtherLink MC-32 (3C527B)			Vendor		Vendor	
3Com EtherLink III (3C509)	Vendor NSTL	Vendor	Yes	Yes		Vendor
3Com EtherLink III (3C509B)			Yes ITL		Vendor ITL	Vendor ITL
3Com EtherLink III-Combo (3C509-COMBO)			Yes		Yes	
3Com EtherLink III-Combo (3C509B-COMBO)			Yes ITL		Vendor ITL	Vendor ITL
3Com EtherLink III-TP (3C509-TP)			Yes		Yes	Vendor
3Com EtherLink III-TP (3C509B-TP)			Yes ITL		Vendor ITL	Vendor ITL
3Com EtherLink III-TPO (3C509-TPO)			Yes		Yes	
3Com EtherLink III-TPO (3C509B-TPO)			Yes ITL		Vendor ITL	Vendor ITL
3Com EtherLink III-EISA (3C579)	Vendor NSTL		Yes ITL	Yes	Vendor ITL	Vendor ITL
3Com EtherLink III-EISA-TP (3C579-TP)			Yes			Vendor
3Com EtherLink III-MCA (3C529)	Vendor NSTL		Yes			Vendor
3Com EtherLink III-MCA-TP (3C529-TP)			Yes		Vendor	
3Com EtherLink III-PCMCIA-Combo (3C589B-Combo)	Vendor* ITL		Vendor*		Vendor ITL	
3Com EtherLink III LAN PC Card (3C589C)	Vendor* ITL		Vendor*		Vendor ITL	
3Com EtherLink III LAN + Modem PC Card (3C562) (LAN ONLY)	Vendor* ITL		Vendor*		Vendor ITL	
3Com TokenLink III (3C619)	Yes NSTL		Yes	Yes		Yes
3Com TokenLink III EISA (3C679)	Yes NSTL	Yes	Yes	Yes		
3Com TokenLink III MCA (3C629)	Yes NSTL		Yes			Yes

<i>Table 21 (Page 2 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
3Com TokenLink III 16/4 PC Card (3C689)			Vendor* ITL		Vendor ITL	
Accton						
Accton EtherCombo-16 (EN1650)			Vendor			Vendor
Accton EtherPair-16 (EN1651)			Vendor			Vendor
Accton EtherCoax-16 (EN1652)			Vendor			
Accton EtherCombo-32 (EN1200)			Vendor			
Advanced Micro Devices, Inc.						
AMD PCnet-32 Ethernet Adapter			Vendor ITL		Vendor ITL	Vendor ITL
AMD PCnet-ISA II Ethernet Adapter			Vendor ITL		Vendor ITL	Vendor ITL
AMD PCnet-PCI Ethernet Adapter			Vendor ITL		Vendor ITL	Vendor ITL
Allied Telesis						
Allied Telesis Ethernet Adapter Card ISA (AT1500-Plus)	Vendor NSTL		Vendor			
Allied Telesis AT1700 Plus ISA	Vendor NSTL		Vendor			
Allied Telesis AT1720 Plus MCA	Vendor NSTL		Vendor			
Artisoft						
Artisoft NodeRunner/SI 2000/C			Yes			
Artisoft NodeRunner/SI 2000/T			Yes		Vendor	
Artisoft NodeRunner/SI 2000/A			Yes		Vendor	
Artisoft NodeRunner/SI 2000M/TC			Yes		Vendor	
Artisoft LANTastic NodeRunner 2000/C			Yes		Vendor	
Artisoft LANTastic NodeRunner 2000/T			Yes			
Artisoft LANTastic NodeRunner 2000/A			Yes		Vendor	
Artisoft LANTastic NodeRunner 2000M/TC			Yes		Vendor	
Asante						
Asante EtherPaC 2000+3			Vendor			
Asante EtherPaC 2000+N			Vendor			
Asante EtherPaC 2000+T			Vendor			
Cabletron Corporation						
Cabletron Ethernet DNI Adapter (E1112)	Vendor NSTL		Yes			
Cabletron Ethernet DNI Adapter (E1119)			Yes			

<i>Table 21 (Page 3 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
Cabletron Ethernet DNI Adapter (E2112)	Vendor NSTL		Yes		Vendor	
Cabletron Ethernet DNI Adapter (E2119)			Yes		Vendor	
Cabletron Ethernet DNI Adapter (E3112)	Vendor NSTL		Yes		Vendor	
Cabletron Ethernet DNI Adapter (E3119)			Yes		Vendor	
Cabletron Token-Ring DNI Adapter (T2015)	Vendor NSTL		Yes		Vendor	
Cabletron Token-Ring DNI Adapter (T3015)	Vendor NSTL		Yes			
CeLAN						
CeLAN FlexLINK - EPCiplus (9910EPCI-B)			Vendor			
Cogent Data Technologies, Inc.						
Cogent eMASTER+ EM960 PCI Ethernet Adapter (EM960C)			Vendor			
Compaq						
Compaq NetFlex-2 ENET-TR Controller	Vendor	Vendor	Vendor	Vendor		
Cray Communications						
Cray Communications ScaNet Network Interface Adapter-ISA	Vendor NSTL		Yes			
Cray Communications ScaNet Network Interface Adapter-MCA	Vendor NSTL		Yes			
Digital Communications Associates						
DCA ClassicBlue MC 4/16 Token-Ring Adapter			Yes			
DCA IRMAtrac EISA			Vendor		Vendor	
DCA IRMAtrac Token-Ring Adapter/Convertible-MCA	Yes NSTL		Vendor			
Digital Equipment Corporation						
Digital EtherWorks 3 Turbo TP (DE204-AA)					Vendor	
Digital EtherWorks Turbo 435 PCI	Vendor NSTL		Vendor			
Digital Semiconductor						
Digital EB40-DECchip 21040 Evaluation Board	Vendor ITL		Vendor ITL		Vendor ITL	Vendor ITL
Digital EB140-DECchip 21140 Evaluation Board	Vendor ITL		Vendor ITL		Vendor ITL	Vendor ITL
D-Link						
D-Link Ethernet Interface Card for the PC XT/AT (DE-220C)			Vendor			

<i>Table 21 (Page 4 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
D-Link Ethernet Interface Card for the PC XT/AT (DE-220CAT)			Vendor			
D-Link Ethernet Interface Card for the PC XT/AT (DE-220CT)			Vendor		Vendor	
D-Link Ethernet Interface Card for the PC XT/AT (DE-220T)			Vendor		Vendor	
D-Link Ethernet Card for EISA bus PC (DE-400)			Vendor		Vendor	
D-Link Ethernet VESA Combo Card (DE-500CAT)			Vendor			
D-Link Ethernet PCI (DE-530CT)			Vendor			
D-Link Token-Ring Adapter for the PC/AT and PS/2 (DT-220)					Vendor	
Eagle Technology						
Eagle Novell NE2000			Yes			
Eagle Novell NE2000T			Yes		Yes	
Eagle Novell NE2000plus			Yes		Yes	
Eagle Novell NE2000Tplus			Yes		Yes	Vendor
Eagle Novell NE2000plus-3			Yes		Yes	Vendor
Eagle EtherXpert EP2000plus	Vendor	Vendor	Yes		Yes	
Eagle EtherXpert EP2000Tplus			Yes		Yes	
Eagle Novell NE3210			Yes		Vendor	Vendor
Eagle EtherXpert EP3210			Yes		Vendor	
Eagle Novell NE/2T					Yes	
Hewlett-Packard						
Hewlett-Packard 27247B			Vendor		Vendor	
Hewlett-Packard PCLAN Adapter/16 PLUS (27252A)			Vendor			
Hewlett-Packard JP2405A					Vendor	
Hewlett-Packard 10/100VG PC LAN ISA Adapter (J2573A)			Vendor			
Hewlett-Packard 10/100VG PC LAN EISA Adapter (J2577A)			Vendor			
Hewlett-Packard 10/100VG PC LAN PCI Adapter (J2585A)			Vendor			
IBM Corporation						
IBM LAN Adapter for Ethernet (48G7169)	Yes	Yes	Yes	Yes	Yes	Yes
IBM LAN Adapter for Ethernet CX (60G615)	Yes		Yes			Yes
IBM LAN Adapter for Ethernet TP (60G0605)	Yes		Yes			Yes

<i>Table 21 (Page 5 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
IBM EtherJet ISA Adapter			<i>Vendor ITL</i>		<i>Vendor ITL</i>	<i>Vendor ITL</i>
IBM EtherJet 10-BASE-T ISA Adapter			<i>Vendor ITL</i>		<i>Vendor ITL</i>	<i>Vendor ITL</i>
IBM Credit Card Adapter for Ethernet 10B2 (0933280)			Yes*			
IBM Credit Card Adapter for Ethernet 10BT (0933290)			Yes*			
IBM Credit Card Adapter II for Ethernet 10B2 (0934330)			Yes*			
IBM Credit Card Adapter II for Ethernet 10BT (0934331)			Yes*			
IBM Adapter/A for Ethernet Networks (6451091)	Yes		Yes			Yes
IBM Adapter/A for Ethernet Twisted-Pair Networks (6451136)			Yes		Vendor	
IBM LAN Adapter/A for Ethernet (48G7171)	Yes		Yes		Vendor	Yes
IBM EtherStreamer MC 32 Adapter (59G9066)			Yes			Vendor
IBM Dual EtherStreamer MC 32 Adapter (73G7136)	Vendor ITL		Yes			
IBM Ethernet Quad-BT PeerMaster Server Adapter (06H5184)	Vendor ITL		Vendor ITL			
IBM Ethernet Quad-B2 PeerMaster Server Adapters (06H6041)	Vendor ITL		Vendor ITL			
IBM Token-Ring Network PC Adapter	Yes		Yes			
IBM Token-Ring Network PC Adapter II	Yes		Yes		Yes	
IBM Token-Ring Network 16/4 Adapter	Yes	Yes	Yes	Yes	Yes	Yes
IBM Token-Ring Network 16/4 ISA-16 Adapter (73G2032)	Yes	Yes	Yes	Yes	Yes	
IBM Token-Ring Network 16/4 Adapter II	Yes		Yes			
IBM Auto 16/4 Token-Ring ISA Adapter (92G7632)	Vendor ITL		Vendor		<i>Vendor</i>	Vendor
IBM 16/4 Busmaster EISA Adapter (1051712)			Yes			
IBM Token-Ring 16/4 Credit Card Adapter (0933462)			Yes*			
IBM Token-Ring 16/4 Credit Card Adapter II (0933931)			Yes*			
IBM PCMCIA Token-Ring Adapter (04H6922)			Yes*			
IBM Token-Ring Network Adapter/A (69X8138)	Yes		Yes		Yes	Yes
IBM Token-Ring Network 16/4 Adapter/A (16F1133)	Yes		Yes		Yes	

<i>Table 21 (Page 6 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
IBM Token-Ring Network 16/4 Adapter/A (74F9410)	Yes		Yes		Yes	Yes
IBM Auto 16/4 Token-Ring MC Adapter (92G7682)			Vendor ITL			Vendor ITL
IBM Token-Ring Network 16/4 Busmaster Server Adapter/A (74F4140)	Yes		Yes			
IBM LANStream MC 16 Adapter (74G0801)			Yes			
IBM LANStream MC 32 Adapter (74G0103)			Yes			
IBM Auto LANStream MC 32 Adapter (60G1592)	Vendor ITL		Yes		Vendor	Vendor
IBM Dual LANStream MC 32 Adapter (73G7137)	Vendor ITL		Yes			Vendor
IBM Auto LANStream PCI Adapter (04H8095)			Vendor ITL		Vendor ITL	Vendor ITL
IBM FDDI Copper Base ISA Adapter			Vendor ITL		Vendor	
IBM FDDI Copper Extender ISA Adapter			Vendor ITL		Vendor	
IBM FDDI Fiber Base ISA Adapter			Vendor ITL		Vendor	
IBM FDDI Fiber Extender ISA Adapter			Vendor ITL		Vendor	
IBM FDDI Copper Base MCA Adapter			Vendor ITL		Vendor	
IBM FDDI Copper Extender MCA Adapter			Vendor ITL		Vendor	
IBM FDDI Fiber Base MCA Adapter			Vendor ITL		Vendor	
IBM FDDI Fiber Extender MCA Adapter			Vendor ITL		Vendor	
IBM Wireless ISA/MCA LAN Adapter	Vendor ITL		Vendor		Vendor	
IBM Wireless PCMCIA LAN Adapter	Vendor* ITL		Vendor*		Vendor	
IBM PC Network Adapter II-Frequency 2	Yes		Yes			
IBM PC Network Adapter II-Frequency 3	Yes		Yes			
IBM PC Network Baseband Adapter	Yes		Yes			
IBM PC Network Broadband Adapter II	Yes		Yes			
IBM PC Network Adapter II/A-Frequency 2	Yes		Yes			
IBM PC Network Adapter II/A-Frequency 3	Yes		Yes			
IBM PC Network Baseband Adapter/A	Yes		Yes			

<i>Table 21 (Page 7 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
IBM PC Network Broadband Adapter II/A	Yes		Yes			
IBM Advanced 3278/79 Emulation Adapter	Yes		Yes			
IBM 3270 Connection, DFT	Yes		Yes			
Intel Corporation						
Intel EtherExpress 16C (PCLA8100)	Vendor NSTL	Vendor	Yes	Yes	Yes	
Intel EtherExpress FlashC (PCLA8105)			Yes	Yes	Yes	
Intel EtherExpress 16 (PCLA8110)			Yes	Yes	Yes	
Intel EtherExpress Flash (PCLA8115)			Yes			
Intel EtherExpress 16TP (PCLA8120)			Yes		Yes	
Intel EtherExpress FlashTP (PCLA8125)			Yes		Yes	
Intel EtherExpress MCA (MCLA8110)			Yes		Yes	
Intel EtherExpress MCATP (MCLA8120)			Yes		Yes	
Intel EtherExpress PRO/10 LAN Adapter (PCLA8200A)	Vendor NSTL		Vendor			
Intel TokenExpress ISA/16S (PCLA8130A)	Vendor NSTL	Vendor	Yes	Yes	Yes	
Intel TokenExpress 16/4 LAN Adapter for EISA (EILA8235)	Vendor NSTL	Vendor	Yes	Yes	Yes	
Intel TokenExpress EISA/32 LAN Adapter (EILA8245)	Vendor NSTL	Vendor	Yes	Yes	Vendor	
Kingston Technology						
Kingston EtherRx PCI Ethernet Adapter (KNE40BT)			Vendor ITL		Vendor ITL	Vendor ITL
Kingston EtherRx LC ISA Combo (KNE2021LC)			Vendor ITL		Vendor ITL	Vendor ITL
Kingston EtherRx LC ISA TP (KNE2000TLC)			Vendor			
Kingston Ethernet PC Card	Vendor* ITL		Vendor*		Vendor ITL	
Kingston TokenRx ISA 16/4 Token-Ring Adapter	Vendor ITL		Vendor		Vendor	
Kingston TokenRx MC 16/4 Token-Ring Adapter	Vendor ITL		Vendor		Vendor	
Kingston TokenRx PCMCIA 16/4 Adapter (KTR-PCM16/4)	Vendor* ITL		Vendor*		Vendor ITL	
LinkSys						
LinkSys Ether16 LAN Card Combo (LNE2000)			Vendor			
LinkSys EtherPCI LAN Card (LNEPCI)			Vendor			
Madge Networks LTD.						

<i>Table 21 (Page 8 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
Madge Smart 16/4 AT PLUS Ringnode (52-03)	Vendor ITL		Vendor		Vendor	
Madge Smart 16/4 ISA Client Plus (22-01)					Vendor	
Madge Smart 16/4 ISA Client PnP Ringnode			Vendor ITL		Vendor ITL	
Madge Smart 16/4 EISA Ringnode (52-08)	Vendor	Vendor	Vendor ITL	Vendor	Vendor ITL	
Madge Smart 16/4 MC Ringnode (54-08)			Vendor		Vendor	
Madge Smart 16/4 MC32 Ringnode (54-09)	Vendor NSTL		Vendor		Vendor	
Madge Smart 16/4 PCMCIA Ringnode (20-00)	Vendor* ITL		Vendor*		Vendor ITL	
Madge Smart 16/4 PCI Ringnode			Vendor ITL		Vendor ITL	
Madge Straight Blue 16/4 ISA (62-01)	Yes	Yes	Yes	Yes		
Madge Straight Blue ISA Plus Blue Box (62-02)			Yes			
Madge Straight Blue MC Blue Box (64-01)			Yes			Yes
Madge Blue+ 16/4 ISA Adapter			Vendor ITL		Vendor ITL	
Madge Smart 100 EISA Ringnode	Vendor ITL		Vendor		Vendor	
Madge Smart 100 AT Ringnode	Vendor ITL		Vendor		Vendor	
NCR Corporation						
NCR StarLAN Token-Ring ISA	Vendor NSTL		Vendor			
NCR StarLAN Token-Ring MCA	Vendor NSTL		Vendor			
NCR Corporation WaveLAN Adapter	Vendor NSTL		Vendor			
Olicom USA, Inc.						
Olicom USA, Inc. ISA 16/4 Adapter (OC-3117)	Vendor NSTL		Yes	Yes	Yes	
Olicom USA, Inc. ISA 16/4 Adapter (OC-3118)			Vendor ITL		Vendor ITL	
Olicom USA, Inc. EISA 16/4 Adapter (OC-3133)			Vendor ITL	Vendor	Yes ITL	
Olicom USA, Inc. EISA/32 Adapter (OC-3135)	Vendor	Vendor	Vendor ITL	Vendor	Vendor ITL	Vendor ITL
Olicom USA, Inc. MCA 16/4 Adapter (OC-3129)			Vendor ITL		Yes ITL	Vendor ITL

<i>Table 21 (Page 9 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
Olicom USA, Inc. Pocket Token-Ring Adapter (OC-3210)			Vendor*		Vendor	
Olicom USA, Inc. Token-Ring PCMCIA Card (OC-3220)	Vendor* ITL		Vendor* ITL		Vendor ITL	
Proteon						
Proteon ProNET/E PCI Ethernet (p1670)			Vendor			
Proteon p1892plus ProNET - 4/16 Plus			Vendor			
Proteon p1392plus ProNET - 4/16 Plus			Vendor		Yes	
Proteon p1990plus ProNET - 4/16 Plus	Vendor	Vendor	Vendor	Vendor	Vendor	
Racal InterLan						
Racal InterLan EtherBlaster TP-8INT (163-3184)			Vendor			
Racal InterLan AT-TP (163-3118)			Vendor			
Racal InterLan NI5210-16 (163-0610)			Vendor			
Racal InterLan ES3210	Vendor	Vendor	Vendor	Vendor		
Racal InterLan ES3210-TP (163-3160)			Vendor			
Racal InterLan MCA (163-3142)			Vendor			
Racal InterLan MCA-TP (163-3143)			Vendor		Vendor	
Racal InterLan PCI T2 (163-3215)			Vendor			
Racal InterLan T/R 16/4 ISA (167-3193)			Vendor		Vendor	
Racal InterLan T/R 16/4 MCA (163-3137)			Vendor		Vendor	
Racore Computer Products, Inc.						
Racore Computer Products, Inc. Token-Ring ISA (M8119)	Vendor NSTL		Yes			
Racore Computer Products, Inc. Token-Ring MC	Vendor NSTL		Yes			
Standard Microsystems Corporation						
SMC EtherCard PLUS (8003EB)	Yes		Yes			
SMC EtherCard PLUS/A (8003E/A)	Yes		Yes			
SMC EtherCard PLUS Elite (8003EP)			Vendor			
SMC EtherCard PLUS Elite 10T (8003WC)			Vendor		Yes	Vendor
SMC EtherCard PLUS Elite 16T (8013WC)			Vendor		Yes	Vendor
SMC EtherCard PLUS Elite 16 (8013EPC)			Vendor			Vendor
SMC EtherCard PLUS Elite 16 Combo (8013EWC)			Vendor		Yes	Vendor
SMC EtherCard PLUS Elite/A (8013EP/A)			Vendor			
SMC EtherCard PLUS Elite 10 T/A (8013WP/A)			Vendor		Yes	
SMC EtherCard Elite 16 Ultra (8216)			Vendor			Vendor

<i>Table 21 (Page 10 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
SMC EtherCard Elite 16C Ultra (8216C)	Vendor NSTL		Vendor	Vendor		Vendor
SMC EtherCard Elite 16T Ultra (8216T)			Vendor		Vendor	Vendor
SMC EtherEZ 10BASE-T ISA (8416T)			Vendor			
SMC EtherCard Elite32C Ultra (82M32C)	Vendor NSTL		Vendor	Vendor		Vendor
SMC EtherPower 10BASE-T PCI Ethernet Adapter (8432T)			Vendor			
SMC EtherPower Combo PCI Ethernet Adapter (8432BT)			Vendor			
SMC EtherPower 10/100 Fast Ethernet PCI Adapter (9332DST)			Vendor			
SMC 9000			Vendor			
SMC TokenCard Elite (8115T)	Vendor ITL NSTL		Vendor		Vendor	
SMC TokenCard Elite/A (8115T/A)			Vendor		Vendor	
SMC TokenCard Elite Master32 (83M32)	Vendor ITL NSTL		Vendor		Vendor	
SysKonnnect, Inc.						
SK-NET FDDI-ISA Network Interface card (SK-5141)	Vendor ITL		Vendor ITL		Vendor ITL	
SK-NET FDDI-EISA Network Interface card (SK-5341)	Vendor ITL		Vendor ITL		Vendor ITL	
SK-NET FDDI-MCA Network Interface card (SK-5241)	Vendor ITL		Vendor ITL		Vendor ITL	
TDK Corporation						
TDK Corporation TDKLAN LAC-CD021 PCMCIA Ethernet Adapter	Vendor* ITL		Vendor* ITL			
Texas Instruments, Inc.						
Texas Instruments, Inc. TokenLite Token-Ring Adapter	Vendor NSTL		Yes			
Thomas-Conrad						
Thomas-Conrad Ethernet PCI (TC5048-T2)			Vendor			
Thomas-Conrad 16/4 Token-Ring Adapter/AT (TC4045)	Vendor ITL		Yes		Vendor	
Thomas-Conrad 16/4 Token-Ring Adapter/MC (TC4046)	Vendor ITL		Yes		Vendor	
Thomas-Conrad Tropic 16/4 Token-Ring Adapter/AT (TC4043)	Yes ITL		Yes		Vendor	

<i>Table 21 (Page 11 of 11). Network Interface Card Support for LAN Systems</i>						
Vendor / Product Name	LS 3.0 NTS/2 7045	LS 3.0 SMP 7045 APAR	LS 4.0 8000	LS 4.0 SMP 8000	DLS 4.0 8000	LAN Distance 1.11
Ungermann-Bass						
Ungermann-Bass NIUpc Adapter	Yes		Yes			
Ungermann-Bass NIUps Adapter	Yes		Yes			
Xircom						
Xircom External Token-Ring Adapter (ET16BU)			Vendor*			
Xircom Pocket Token Ring Adapter III (PT3-16CTP)			Vendor*		Vendor	
Xircom CreditCard Token Ring Adapter (CT-16CTP)			Vendor*			
Xircom Pocket Ethernet Adapter III (PE3-10BT)			Vendor*			
Xircom Pocket Ethernet Adapter III (PE3-10BC)			Vendor*			
Xircom Pocket Ethernet Adapter III (PE3-10BX)			Vendor*			
Xircom CreditCard Ethernet Adapter (CE-10BC)			Vendor*			
Xircom PS-CreditCard Ethernet Adapter (PS-CE2-10BT)			Vendor*			

All trademarks contained herein are the property of their respective trademark owners.

DOS LAN Services 4.0 together with LAN Support Program supports a different list of network adapter cards. Please refer to LAN Server V4.0 documentation for more information.

E.3.1 Support for Additional Drivers

Additional device drivers are shipped with NTS/2 on the Additional Network Adapter Support diskette. These additional device drivers will not be stored on the code server when loading the LAPS diskette image as described in 16.1.2, "Loading LAN Transport System Diskette Image(s) with LAPSDISK" on page 382.

You might also have additional device drivers from other sources that you want to add to the LAPS image in order to support other drivers.

* This adapter card is supported on clients only.

Below is an example which will show you what needs to be done:

1. Update of the code server LAPS image for the IBM Token-Ring Network 16/4 Credit Card Adapter.
2. Create a valid LAPS response file on the code server.
3. Enable THINLAPS to transfer the IBMTOKCS.OS2 driver according to its associated .NIF file, IBMTOKCS.NIF, to the LTS diskette.

Note

- The IBMTOKCS.OS2 driver, IBMTOKCS.NIF file and associated *.MSG message files are located on the on the IBM Token-Ring 16/4 Credit Card Adapter Diagnostics Diskette. Make sure to use the latest version of this diskette.
- The following steps are performed on the code server, assuming that the CID directory structure is on the D: drive.

The updating of the code server is slightly changed between NTS/2 LAPS and MPTS LAPS, see below:

Update of the code server NTS/2 LAPS diskette image:.

1. Make a backup copy of MACS.ZIP file stored on code server.
`COPY D:\cid\img\laps\ibmcommacs\macs.zip *.old`
2. Make a backup copy of IBMCOM.ZIP file stored on code server.
`COPY D:\cid\img\laps\ibmcom\ibmcom.zip *.old`
3. Insert the IBM Token-Ring 16/4 Credit Card Adapter Diagnostics Diskette in drive A:.
4. Add additional driver to LAPS diskette image using PKZIP2.
`PKZIP2 D:\cid\img\laps\ibmcommacs\macs.zip A:\ibmtokcs*.*`
`PKZIP2 \cid\img\laps\ibmcom\ibmcom.zip A:\ltg*.*`

Update of the code server MPTS LAPS diskette image:.

1. Insert the IBM Token-Ring 16/4 Credit Card Adapter Diagnostics Diskette in drive A:.
2. Copy the additional driver to LAPS diskette image.
`COPY A:\ibmtokcs*.* D:\cid\img\laps\ibmcommacs`
`COPY A:\ltg*.* D:\cid\img\laps\ibmcom`

The NIF file and the adapter driver files shall be copied to the IBMCOMMACS directory, as in the example above. The message files shall be copied to the IBMCOM directory as in the example. The additional network adapter drivers and associated files can be in packed or unpacked format. If they are in packed format, they must have been packed by the PKZIP2 program and they must have a file name extension of .ZIP.

To find out for other adapters which files needs to be copied browse the NIF file and look at the KEYWORDS CopyFile (Optional) and Name. Make sure to read the IBMCOMMACSREADMAC.TXT file on a system, with MPTS installed, for special considerations for some adapters.

Create LAPS response file on code server:

The LAPSRSP utility will create a LAPS response file based on a valid PROTOCOL.INI. The following steps will generate a valid NTS/2 PROTOCOL.INI.

In 3.2.4, "MPTS Response File" on page 58 there is a detailed description on how to create a valid PROTOCOL.INI covering both NTS/2 and MPTS LAPS. The following steps will generate a valid NTS/2 PROTOCOL.INI. (Some additional steps are needed for MPTS):

1. Start LAPS on the code server or on any other workstation. Make a backup copy of your current PROTOCOL.INI and CONFIG.SYS because the following steps will generate a new PROTOCOL.INI and modify CONFIG.SYS.

Note

The IBM Token-Ring 16/4 Credit Card Adapter does not need to be installed in this workstation.

Execute:

```
COPY C:\config.sys *.bak
COPY C:\ibmcom\protocol.ini *.bak
C:\ibmcom\laps
```

and press Enter

2. Select **INSTALL** from LAPS main menu.
3. Insert the IBM Token-Ring 16/4 Credit Card Adapter Diagnostics Diskette in drive A:.
4. Specify **Source of NIF file** = A:.

5. Select **CONFIGURE**.
6. Select **Configure LAN transport**.
7. The IBM Token-Ring 16/4 Credit Card Adapter now appears in the list of available Network Adapters.
8. Remove current Protocols and Network Adapters from current configuration.
9. Select **IBM Token-Ring 16/4 Credit Card Adapter** and add it to the current configuration.
10. Select **IBM OS/2 NetBIOS protocol** and add it to the current configuration.
11. Select **IBM IEEE 802.2 protocol** and add it to current configuration if you intend later on to install ES 1.0, LS 3.0, CM/2 or any other application running on top of IEEE 802.2 interface. If you do this and want to use the newly created PROTOCOL.INI as input for THINLAPS later, remember to remove these protocol definitions before running
12. Select **IBM Token-Ring 16/4 Credit Card Adapter** and **EDIT**.
13. Fill in the values for at least Interrupt Level and Shared RAM. 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. If you want to see which values are assigned to the Credit Card adapter by the system, boot the machine with the IBM Token-Ring 16/4 Credit Card Adapter Diagnostics Diskette and run the adapter diagnostics. On an installed ThinkPad system, you can check these values with the ThinkPad utilities. If you need information about the Interrupt Level and other questions about ThinkPad machines, please refer to *ThinkPad Systems*, GG24-4297. Specify the Ring Speed according to your network.
14. Select **OK**.
15. Select **EXIT**.
16. Select drive on which CONFIG.SYS should be updated.
17. Select **CONTINUE**.
18. Select **OK** for successful update of CONFIG.SYS.
19. Exit LAPS.
20. Save the PROTOCOL.INI created. You may want to use it to recreate response files for machines with PCMCIA systems and for the execution of THINLAPS.

COPY C:IBMCOMP/PROTOCOL.INI PROTOCOL.TPD
21. Copy the original versions of CONFIG.SYS and PROTOCOL.INI back.

```
COPY C:\config.bak *.sys
COPY C:\ibmcom\protocol.bak *.ini
```

The following figure shows a sample PROTOCOL.INI created by NTS/2 LAPS for the IBM Token-Ring 16/4 Credit Card Adapter including 802.2 support. A valid MPTS file would look almost the same. Please remember that the settings for the adapter may be different in your case.

IBM Token-Ring 16/4 Credit Card Adapter

[PROT_MAN]

DRIVERNAME = PROTMAN\$

[IBMLXCFG]

landd_nif = landd.nif

netbeui_nif = netbeui.nif

IBMTOKCS_nif = IBMTOKCS.nif

[landd_nif]

DriverName = LANDD\$

Bindings = IBMTOKCS_nif

ETHERAND_TYPE = "I"

SYSTEM_KEY = 0x0

OPEN_OPTIONS = 0x2000

TRACE = 0x0

LINKS = 8

MAX_SAPS = 3

MAX_G_SAPS = 0

USERS = 3

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 = 800

Figure 112 (Part 1 of 2). NTS/2 LAPS PROTOCOL.INI

```

[netbeui_nif]

DriverName = netbeui$
Bindings = IBMTOKCS_nif
ETHERAND_TYPE = "I"
USEADDRREV = "YES"
OS2TRACEMASK = 0x0
SESSIONS = 40
NCBS = 95
NAMES = 21
SELECTORS = 5
USEMAXDATAGRAM = "NO"
ADAPTRATE = 1000
WINDOWERRORS = 0
MAXDATARCV = 4168
T1 = 60000
T1 = 10000
T2 = 5000
MAXIN = 1
MAXOUT = 1
NETBIOS_TIMEOUT = 500
NETBIOS_RETRIES = 8
NAMECACHE = 0
PIGGYBACKACKS = 1
DATAGRAMPACKETS = 2
PACKETS = 350
LOOPPACKETS = 1
PIPELINE = 5
MAXTRANSMITS = 6
MINTRANSMITS = 2
DLCRETRIES = 5
FCPRIORITY = 5
NETFLAGS = 0x0

[IBMTOKCS_nif]

DriverName = IBMTOK$
ADAPTER = "PRIMARY"
MAXTRANSMITS = 6
RECVBUFS = 2
RECVBUFSIZE = 256
XMITBUFS = 1
INTERRUPT = 9
PCMCIA
RAM = 0xD800
RINGSPEED = 4
RAMSIZE = 16
MMIO = 0xD000

```

Figure 112 (Part 2 of 2). NTS/2 LAPS PROTOCOL.INI

This PROTOCOL.INI file can now be used as an input file for LAPSRSP.EXE in order to create a valid NTS/2 LAPS response file on the code server.

**LAPSRSP C:\ibmcom\protocol.tpd D:\cid\rsp\laps\trcca.rsp
/T:c: /U:new /I:product**

The LAPSRSP command can be used in the same way for MPTS, but can also be used with more parameters. Please see 3.2.4, "MPTS Response File" on page 58 for more information.

Note on TRCCA.RSP

TRCCA.RSP is a valid response file for LAPS redirected installation on AT-bus workstations equipped with the IBM Token-Ring 16/4 Credit Card Adapter.

The following figure shows a sample NTS/2 LAPS response file TRCCA.RSP created by LAPSRSP.EXE for the IBM Token-Ring 16/4 Credit Card Adapter:

```

INST_SECTION = (
  TARGET = c:
  UPGRADE_LEVEL = new
  INSTALL = product
)

PROTOCOL = (
[PROT_MAN]

  DRIVERNAME = PROTMAN$

[IBMLXCFG]

  landd_nif = landd.nif
  netbeui_nif = netbeui.nif
  IBMTOKCS_nif = IBMTOKCS.nif

[landd_nif]

  DriverName = LANDD$
  Bindings = IBMTOKCS_nif
  ETHERAND_TYPE = "I"
  SYSTEM_KEY = 0x0
  OPEN_OPTIONS = 0x2000
  TRACE = 0x0
  LINKS = 8
  MAX_SAPS = 3
  MAX_G_SAPS = 0
  USERS = 3
  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 = 800

```

Figure 113 (Part 1 of 2). Sample NTS/2 LAPS Response File TRCCA.RSP for IBM Token-Ring 16/4 Credit Card Adapter

```

[netbeui_nif]

    DriverName = netbeui$
    Bindings = IBMTOKCS_nif
    ETHERAND_TYPE = "I"
    USEADDRREV = "YES"
    OS2TRACEMASK = 0x0
    SESSIONS = 40
    NCBS = 95
    NAMES = 21
    SELECTORS = 5
    USEMAXDATAGRAM = "NO"
    ADAPTRATE = 1000
    WINDOWERRORS = 0
    MAXDATARCV = 4168
    T1 = 60000
    T1 = 10000
    T2 = 5000
    MAXIN = 1
    MAXOUT = 1
    NETBIOS_TIMEOUT = 500
    NETBIOS_RETRIES = 8
    NAMECACHE = 0
    PIGGYBACKACKS = 1
    DATAGRAMPACKETS = 2
    PACKETS = 350
    LOOPPACKETS = 1
    PIPELINE = 5
    MAXTRANSMITS = 6
    MINTRANSMITS = 2
    DLCRETRIES = 5
    FCPRIORITY = 5
    NETFLAGS = 0x0

[IBMTOKCS_nif]

    DriverName = IBMTOK$
    ADAPTER = "PRIMARY"
    MAXTRANSMITS = 6
    RECVBUFS = 2
    RECVBUFSIZE = 256
    XMITBUFS = 1
    INTERRUPT = 9
    PCMCIA
    RAM = 0xD800
    RINGSPEED = 4
    RAMSIZE = 16
    MMIO = 0xD000

)

```

Figure 113 (Part 2 of 2). Sample NTS/2 LAPS Response File TRCCA.RSP for IBM Token-Ring 16/4 Credit Card Adapter

Run THINLAPS to transfer network driver to LTS diskette:

The LAPS diskette image is now updated on the code server and THINLAPS can be executed to transfer NetBIOS and the IBM Token-Ring 16/4 Credit Card Adapter network driver to the LTS diskette. It is a good idea to use the PROTOCOL.INI file created in the preceding step as an input file to get all

adapter definitions correctly. Though you might want to reduce the number of protocols defined because only NetBIOS is needed.

D:\cidimglaps\THINLAPS D:\cidimglaps A: IBMTOKCS.NIF /P:C:\IBMCOM\PROTOCOL.TPD

The LTS diskette can now be used on PCMCIA-bus machine equipped with the IBM Token-Ring 16/4 Credit Card Adapter. Do not forget to add the files needed for the PCMCIA support as described in I.3, "PCMCIA and CID" on page 576.

Appendix F. Create Environment Variables Program Description

The Create Environment Variables Program (CRENVVAR.EXE) is used in the installation procedures for Novell NetWare requester and TCP/IP V2.0.

F.1 How to Use CRENVVAR.EXE

This program prompts for environment variables. It lets the user define the name of the variable and type the prompt string, which will be shown when the program requests user input. The program requires input and will repeat the same prompt, until the user enters *any* data. The name of the variable and the entered data are composed together to form a valid OS/2 "SET" statement which is stored in a CMD procedure called ENV_VARS.CMD. The program deletes this file upon program entry. So if the program crashes or receives invalid data, the file ENV_VARS.CMD does not exist. By executing the ENV_VARS.CMD after CRENVVAR.EXE has finished, the environment variable becomes part of the current OS/2 environment.

Two parameters are valid for CRENVVAR. Both parameters are required and used in conjunction. The program always searches for a pair, thereby interpreting the command line input from left to right. The two parameters don't need to be in a specific order. A request is only put out if both parameters are present.

No blanks are allowed between parameter and data and parameter and double-quoted string data (within a string blanks are allowed). Parameters are separated by one or more blanks.

More than one variable can be set by one program execution. After the first pair has been interpreted and executed, the program continues with the next set (if available) thereby moving from left to right until all parameters are processed.

Issuing **CRENVVAR ?** outputs a brief explanation of the function and its usage. (See the program list below for its content.)

The two parameters are:

- /V:** With this parameter the user defines the name of the new or replaceable environment variable. Any name valid as environment variable can be used. If the name already exists, its value will be replaced.
- /P:** This parameter defines the content of the prompt string. This is the string the user will see, when being asked to enter the data for a specific variable. If the string contains blanks as word separator, the string must be embedded by ""(double quotes). No colon is needed at the end of the string. It is automatically added by the program.

F.2 Samples with CRENVVAR.EXE

The first example shows the prompting for one variable. The /v: and /p: parameters could be in any order.

The program call:

```
CRENVVAR /P:"Please enter your workstation name" /V:WSNAME
```

or

```
CRENVVAR /v:WSNAME /p:"Please enter your workstation name"
```

would both result in the following execution:

```
(WSNAME) Please enter your workstation name:
(an assumed input from the user could be mywps)
```

which results in creation of ENV_VARS.CMD file with the following content:

```
SET WSNAME=MYWPS
```

The following example issues two prompts for two variables:

```
Program call:
CRENVVAR /V:WSN /P:"WS Name" v:WSA /P:"WS address"
```

F.3 Make File, DEF File and Source Code for CRENVVAR.EXE

The following section shows all information needed to create the environment variable prompter program.

F.3.1 Make File for CRENVVAR C Routine

Compiler control file used when compiling and link editing the CRENVVAR.EXE, the source for which is listed later.

```
crenvvar.exe: crenvvar.obj crenvvar.def
    link crenvvar /A:16,crenvvar,crenvvar,llibc+os2 /N0d /MAP,crenvvar.def;

crenvvar.obj: crenvvar.c
    cl /c /Od /Zi /Zp /A1fu /W3 /G2s /Gc crenvvar.c
```

F.3.2 CRENVVAR.DEF Define File

```
NAME      CRENVVAR      WINDOWCOMPAT

DESCRIPTION  'Create Environment Variables'

STUB      'OS2STUB.EXE'

DATA      MULTIPLE

STACKSIZE  4092
HEAPSIZE   4092

PROTMODE
```

F.3.3 C Source for CRENVVAR.C

This module is the main routine for the create environment variables program.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <os2.h>

/*****
/* prototypes and */
/* global variables */
*****/
SHORT cdecl reqputenv(PCHAR,PCHAR);

static CHAR Buffer[1000];
static CHAR Env_Vars[] = "ENV_VARS.CMD";
static FILE *myfile;

/*****
/* */
/* Start of main program */
/* */
*****/
void cdecl main(argc, argv)
```

```

        int      argc;
        char     *argv[];
    {
        /*****
        /*  local variables
        *****/
        static CHAR    Kwd1[] = "/V:";
        static CHAR    Kwd2[] = "/P:";
        PCHAR          EnvVar;
        PCHAR          PromptString;
        int            i;

        /*****
        /*  description of program
        *****/
        if (argc > 1) {
            if (!strcmp(argv[1], "?"))
            { printf(
              "  Create Environment Variables\n"
              "  _____\n"
              "\n"
              "  Syntax:\n"
              "\n"
              /*
              "  CRENVVAR /V:EnvVariableName /P:"PromptString"\n"
              */
              "
              "
              "  CRENVVAR  → [ /V:envvariablename ] → \n"
              "                  [ /P:"PromptString"\n ] → \n"
              "\n");
              printf(
              "
              EnvVariableName  Name of environment variable to be created\n"
              PromptString      Prompt string layout \n"
              "\n");
              printf(
              "
              This program always expects a pair ( /v: /p:\n"
              "in any order. It prompts as soon, as a /v: and /p: \n"
              "are detected. The result is stored in %s\n"
              "If this file does not exist, nothing was retrieved.\n"
              "\n", Env_Vars);
              printf(
              "\n"
              "  ITS0 Boca Raton, Florida\n");

              exit(0);
            }

            if (argc == 1) {return;}          /* do nothing if user asked for nothing */

            /*****
            /*  generalised section, initialization
            *****/

            Buffer[0]=' \0';
            i=0;
            myfile=NULL;
            remove(Env_Vars);

            /*****
            /*  parsing the input and execute each pair of parms
            *****/
            do {

```

```

EnvVar = NULL;
PromptString = NULL;

do {
    i++;
    if (strlen(argv[i]) >= sizeof(Kwd1)) {
        if (!memcmp(argv[i], Kwd1, sizeof(Kwd1) - 1)) {
            strupr(EnvVar = &argv[i][sizeof(Kwd1) - 1]);
            continue;
        }
    }
    if (strlen(argv[i]) >= sizeof(Kwd2)) {
        if (!memcmp(argv[i], Kwd2, sizeof(Kwd2) - 1)) {
            PromptString = &argv[i][sizeof(Kwd2) - 1];
            continue;
        }
    }
}
/*****
/* ok, something unknown had been entered, we quit */
/*****
printf("Unknown keyword encountered, '%s', Program ended.\n",argv[i]);
return;
} while (((PromptString == NULL) || (EnvVar == NULL)) && (i < (argc-1)));
if ((PromptString == NULL) || (EnvVar == NULL)) { return;}

/*****
/* we found a pear(...) lets execute it and exit if an error occurred */
/*****
if (regputenv(EnvVar,PromptString) != 0) {break;}
} while (i < (argc-1));

/*****
/* we did it, lets call it a day */
/*****
if (myfile != NULL) {fclose(myfile);} /* be nice to a friend (OS/2) */
return;
}

/*****
/* request the variable input and put it in the "Env_Vars" file */
/*****
SHORT cdec1 regputenv(PCHAR EnvVar , PCHAR PromptString)
{
CHAR vardata[256]; /* input string for user */
SHORT rc;
printf("\n"); /* grab a new line */

do {
    /* repeat until something has been entered */
    printf("(%s) %s:", EnvVar, PromptString); /* prompt the user */
    gets(vardata); /* get the input from the user */
} while (strlen(vardata) == 0);

strupr(vardata); /* make it all uppercase */
if (myfile == NULL) { /* if not open, open the file */
    myfile=fopen(Env_Vars,"w");
}

if (myfile == NULL) { /* if open went wrong, tell the world and quit */
    printf("trouble with fopen");
    return(1);
}

sprintf(Buffer,"SET %s=%s\n",EnvVar,vardata); /* store the command */
rc=fwrite(Buffer,strlen(Buffer),1,myfile); /* and write it */

```

```
    if (rc == 0) {                /* if 0 bytes written, we have an error */
        printf("trouble with fwrite");
        return(1);
    }

    return(0);                    /* we are done with this one */
}
```

Appendix G. Use of Other Code Servers

This appendix will describe the use of CID with three products/scenarios: LAN File Services/ESA, LAN Resource Extension and Services/VM, and IBM Client Access for AS/400, previously known as IBM PC Support/400. For detailed information please refer to the document *Workstation LAN File Services/VM, LAN Resource Extension and Services/VM, AS/400*, GG24-4073-00. The figure below shows a descriptive view of the CID installation using host DASD.

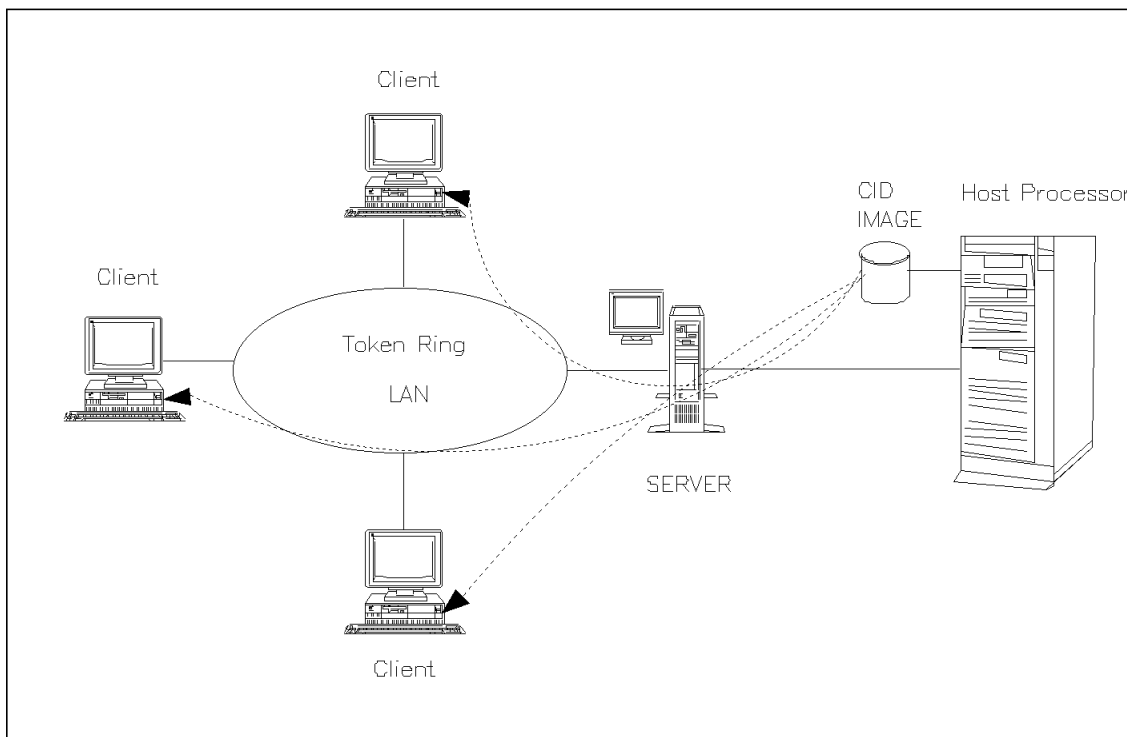


Figure 114. CID Installation Using Host DASD

G.1 LAN File Services/ESA

LAN File Services/ESA (LFS/ESA) brings together computing environments that were previously separate entities.

Historically, Local Area Network (LAN) data and VM data have been stored as separate entities. LAN data has been saved on PC-based LAN servers or on users' local disk drives, while VM data resided on large capacity System/370* or System/390* DASD. Interaction between the two environments consisted of occasional uploads and downloads of files to and from the VM system, but even with this, separate copies of data were still maintained.

With LFS/ESA, many of the barriers between the VM host environment and the LAN are removed and the strengths of each environment complement those of the other. As a result, new function and capacity are added to each of these environments.

LAN File Services/ESA:

- Uses VM DASD to provide file sharing services through LAN servers to workstation users.
- Allows file sharing across multiple LANs.
- Provides its services transparently.
- Allows sharing between OS/2 LAN server requesters and Network File System (NFS) clients.

Some of the fundamental ideas in setting up and administering a LFS/ESA system:

- In an OS/2 LAN environment, LFS/ESA acts as an extension of the OS/2 LAN server.
- LFS/ESA uses configuration files. Some of the files reside on the VM system and some reside on each OS/2 LAN Server system that acts as a front end processor for LFS/ESA. These files are read once each time LFS/ESA is started. Every time LFS/ESA is started it uses the current values in the configuration files.
- Temporary changes can be made via administrative commands to LFS/ESA while LFS/ESA is running. It is not always necessary to stop LFS/ESA to make important changes.

- Almost all of the administration of LFS/ESA is done from an administration virtual machine other than the server virtual machine.

LFS/ESA in a OS/2 environment has three parts:

1. An administration user ID on the VM system
2. A Server Message Block (SMB) protocol server on the VM system OS/2 LAN Server environments use the Microsoft Server Message Block (SMB) server protocols over lower-layer NetBIOS communications protocols.
3. A corresponding client in a PS/2* OS/2 running LAN Server that acts as a "front end processor" to the VM system.

In an OS/2 environment, LFS/ESA can use two different connectivity methods:

1. CLAW (Common Link Access to Workstations)
2. VM PWSCS (Programmable Workstations Communications Services)

LAN topologies supported in the OS/2 environment include token-ring and Ethernet.

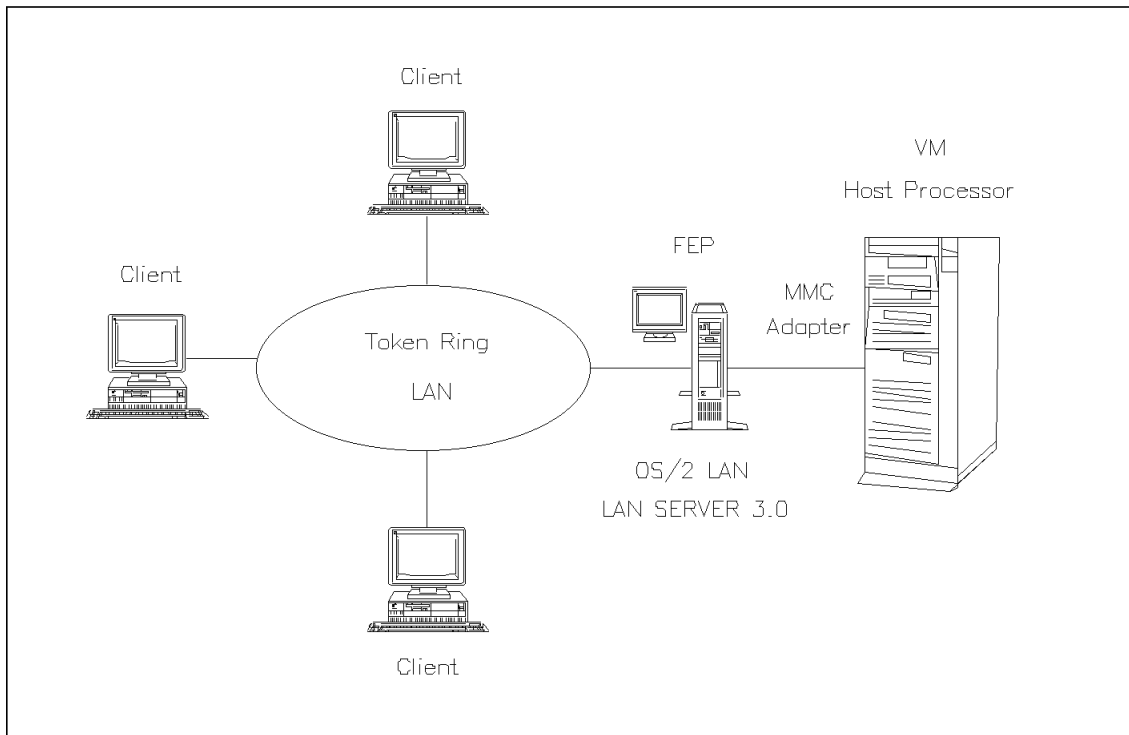


Figure 115. CID Installation Using Host DASD

G.1.1 CID Installation

The Figure 115 on page 554 displays a typical scenario. The OS/2 LAN server with help of the LFS/ESA Front End Processor (FEP) defines the alias for the host DASD. The SRVIFS server is installed on the same physical workstation as OS/2 LAN server and the FEP. The CID directory structure can thereby be kept on the host DASD. The SRVIFS server maintains the LCU functions with respect to the clients.

G.2 LAN Resource Extension and Services/VM

The LAN Resource Extension and Services/VM (LANRES), Program Number 5684-142, is an IBM product that provides services to NetWare clients by using virtual machine (VM) resources.

LANRES gives NetWare clients more disk storage space by making S/390* and S/370* direct access storage devices (DASD) accessible to the NetWare servers. It also puts VM system printers at the NetWare client's disposal.

LANRES/VM extends the NetWare environment to include the S/390 and S/370 host. Because it does this transparently, NetWare clients are unaware of the LAN-host interaction. They retain all the advantages of working in a LAN environment but receive use of VM large-capacity DASD and high-speed printers.

LANRES/VM also provides a data distribution service to help with change management. Authorized VM users can send data to, and retrieve data from, the NetWare server. They can list server files and directories, and create and delete server files.

Besides making VM resources available to NetWare servers and clients, LANRES/VM makes LAN printer resources available to VM users. For example, VM users can now send PostScript** files to a PostScript printer on the LAN.

In addition to disk and print serving, LANRES/VM lets you move your LAN administration to VM, where tasks can be automated and where multiple LANS can be centrally administered from the host.

REXX programs provided with the product or written by users can be combined to perform new functions for LANRES/VM.

In summary, these are the services that LANRES/VM provides:

- Disk serving
- Data distribution
- Print serving
 - LAN-to-host printing
 - Host-to-LAN printing
- LAN administration

The intent of LANRES/VM is to retain the advantages of LANs for workstation responsiveness, availability, and inter-workstation communication, while bringing to the LAN such System/390 and System/370 resources as large capacity DASD, high speed printers, and wide area networking. LANRES/VM also makes LAN printer resources available to VM users.

In addition, LANRES/VM provides facilities for LAN administration and data distribution. With LANRES/VM, VM users can handle administration problems and changes, as well as manage NetWare server files and directories, from a central location. By requiring a NetWare server logon and password, LANRES/VM limits the range of these users' activities to those defined by their NetWare security privileges.

LANRES/VM services complement existing NetWare functions. Your installation can decide which LANRES/VM services to use; they can be used in any combination. NetWare servers and clients can still use local disks and printers. You can use the NetWare SYSCON utility for administration in conjunction with LANRES/VM administration functions.

One VM system can provide concurrent services for multiple NetWare servers. To do this you need to have multiple VM service machines: one or more for LAN-to-host printing, one or more for disk serving, and one for host-to-LAN printing for each NetWare server that is attached to VM. The VM system can also provide services to NetWare servers that are not directly connected to it if they are accessible to a NetWare server that is directly connected. Each VM user doing administration, data distribution, or host-to-LAN printing works with one server at a time and can switch easily from server to server.

The LANRES/VM data distribution, LAN administration, and host-to-LAN printing services are conversational monitor system (CMS) programs. They consist of line-oriented commands, so they can be used in REXX programs to automate procedures. They can also be used from any VM-supported terminal. And they can be used from terminals connected through the VM/Pass-Through Facility or the Virtual Telecommunications Access Method; this feature lets administrators control LANS connected to other VM systems in a wide area network.

The LANRES/VM disk serving and LAN-to-host printing functions are transparent to NetWare clients. Clients use the same commands as always, and if the disk or printer they specify happens to be a host resource, LANRES/VM provides the function needed to use that resource.

All LANRES/VM services are available to any supported NetWare clients. This includes full support for DOS, Microsoft Windows, OS/2, Macintosh, and UNIX clients. The figure below shows a descriptive view of the services provided by LANRES.

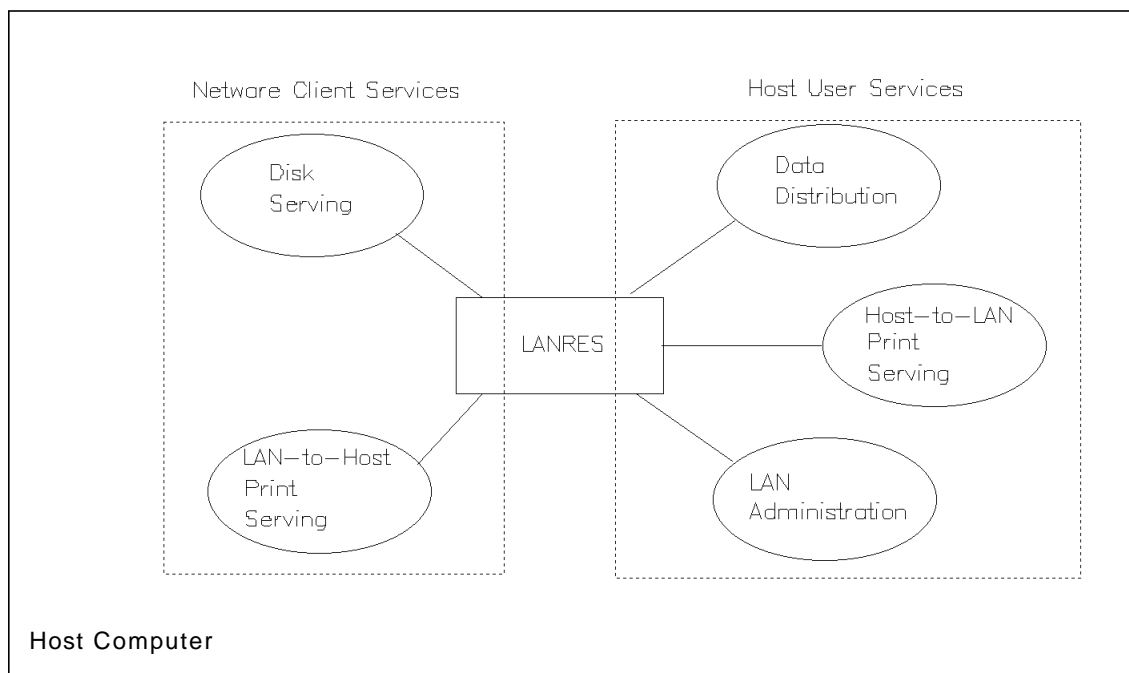


Figure 116. Services Provided by LANRES

G.2.1 CID Installation

Figure 117 on page 558 displays a typical scenario. The Novell NetWare server with help of LANRES server program defines the volume for the host DASD. The CID directory can be kept there. The Novell NetWare server maintains the CID installations with respect to the clients.

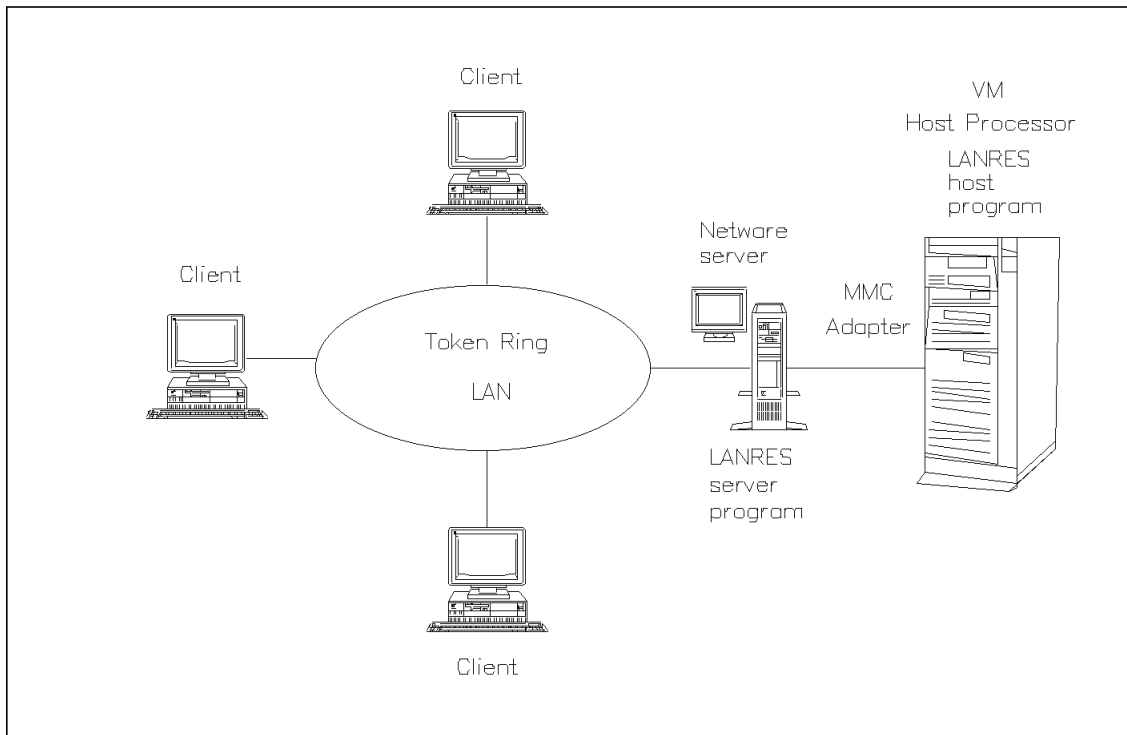


Figure 117. CID Installation Using Host DASD

G.3 IBM Client Access for AS/400

Client Access for AS/400 is the premier client/server offering for the AS/400 system and the premier cooperative processing application enabler for the AS/400 system.

Client Access provides similar client/server functions (such as file serving and resource sharing) as other client/server products (OS/2 LAN Server, Novell NetWare, etc.). However, the AS/400 system and Client Access support are best promoted as a "High Function Server" to customers who have requirements that cannot be satisfied by commodity PC servers. Some of the strengths of the AS/400 system as a High Function Server are its powerful, built-in database, comprehensive security, advanced networking LAN/WAN transparency, and strong management of local and remote

computing environments. Client Access has been designed to bring this power of the AS/400 system to the desktop. Customers using this combination can:

- Take advantage of any of the thousands of available PC applications and centralize and share their data transparently on an AS/400 system, and at the same time capitalize on the increased power and sophistication available in an OS/400 host environment.
- Select any of the thousands of available AS/400 applications that might be right for their business and use programmable as well as non-programmable workstations.

One of Client Access' many additional strengths is its PC Software Update capability. By combining the systems management capabilities of the OS/400 with the Client Access function, customers can update software on all the PCs in their network from one single location, transparently to end users.

The AS/400 system is positioned as a "Cooperative Applications Server". Client Access enables cooperative application development. The customers can now fully develop cooperative applications to the AS/400 system within a Windows environment. These exciting new application enhancements should be conveyed to both AS/400 programmers as well as PC programmers as there are many new enablers being provided for both kinds of developers.

Client Access for AS/400 provides a flexible set of cooperative processing functions for customers who need to take advantage of AS/400 data, programs, and resources from OS/2, Windows or DOS workstations. Client Access integrates the strengths of the AS/400 environment with the power and ease-of-use of the programmable workstation by providing:

- A comprehensive set of Application Programming Interfaces
- Database access via SQL
- File transfer
- File serving
- Print serving
- An integrated access to both AS/400 applications and PC applications
- Automatic update of programmable workstation software
- Centralized systems administration

G.3.1 Connectivity Alternatives

Client Access provides the following connections for DOS and Windows based systems to the AS/400 system:

- IBM Token-Ring Network (direct and 5494 Remote Controller)
- Twinaxial (local, 5394 and 5494 Remote Controllers)
- Asynchronous (via direct attach through IBM or ROLM** CBX and remote dialup)
- SDLC
- Ethernet (Native or 8209 Bridge)
- 3174 Establishment Controller APPN*, Peer Communication LIC, and Configuration Support-C

Only twinaxial, token-ring, Ethernet, and SDLC connections are available for DBCS.

The IBM OS/2 Communications Manager provides the following APPC connections to the AS/400 system:

- IBM Token-Ring Network (direct and 5494 Remote Controller)
- Twinaxial (local, 5394 and 5494 Remote Controllers)
- X.25
- SDLC
- Ethernet (Native or 8209 Bridge)

G.3.2 CID Installation

The Figure 118 on page 561 displays a typical scenario. Client Access identifies the shared folders. The SRVIFS server executes on the same physical workstation as Client Access. The SRVIFS server defines the aliases for the shared folders available via Client Access, so the CID directory structure can be kept on the AS/400 DASD. The SRVIFS server maintains the LCU functionalities towards the clients.

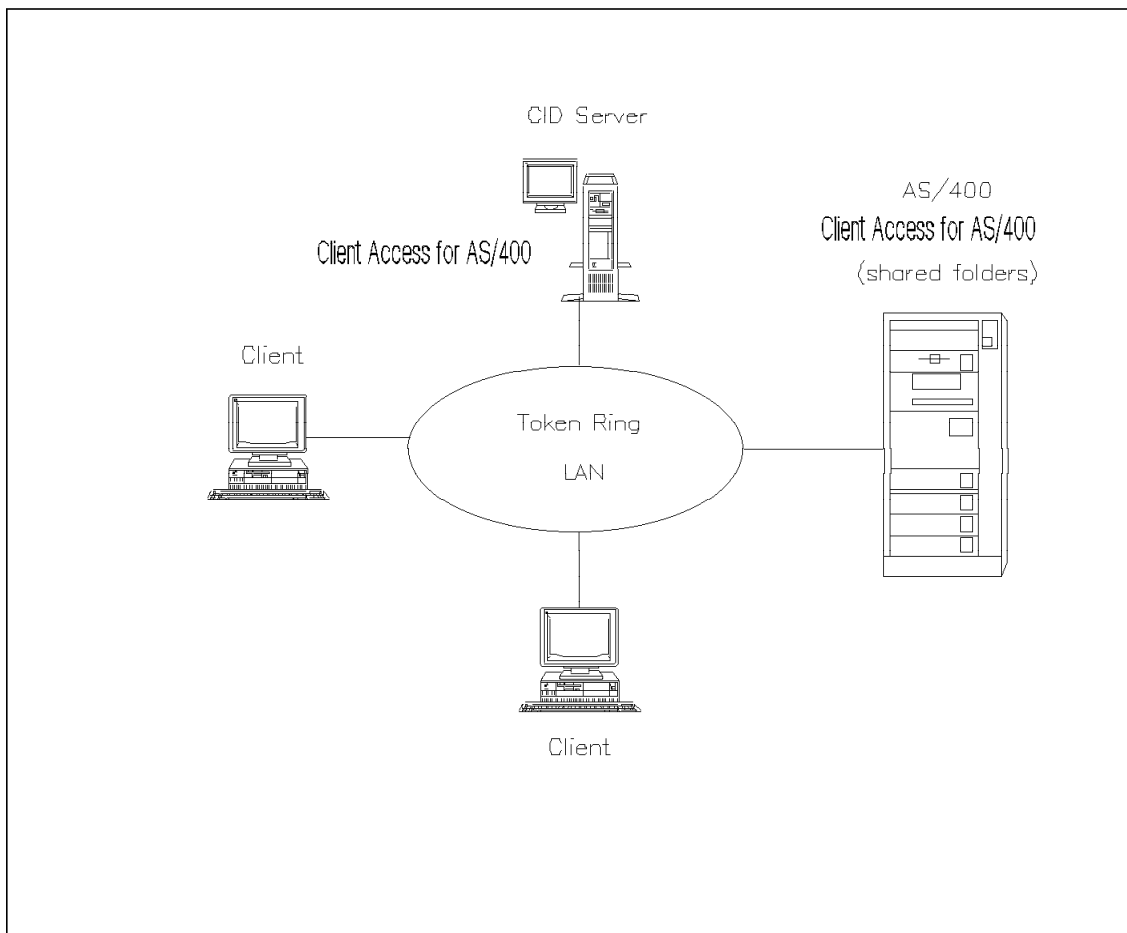


Figure 118. CID Installation Using Host DASD

Appendix H. Sample Code Diskette/CDROM

Multiple DEFAULT.CMD

For each type of LAN Transport environment there is a subdirectory and in each there is a CONNECT.CMD (except for NVMD/2). These CONNECT.CMDs are different. The installation queues are tuned to be good working examples for each type of server and to install what is needed for the specific environment. The sample batches were updated for the newer software releases. The previous versions are stored in the **GG244295IMAGES** directory.

D:		
—EZ2INST		IBM Library Reader subdirectory
—BOOKS		Earlier versions of CID redbooks
—GG244295.B00		GG24-4295-00
—IMAGES		Directories with old sample diskettes
—GG244295		GG24-4295-00 diskette content
—MISC		Miscellaneous files
—RESPONSE.INF		
—PCSREF.INF		
—NETWARE		NETWARE subdirectory
—CLIENT1.CMD		
—CLIENT1.RSP		
—DEFAULT.CMD		
—GETREXX.CMD		
—LAPSRSP.RSP		
—NWDELETE.CMD		
—NWICON.CMD		
—NWINST.CMD		
—NWPREP.CMD		
—NWSEED.CMD		
—OS2V20.RSP		
—STARTNW.CMD		
—STARTRFI.CMD		

Figure 119 (Part 1 of 8). The Sample Code CDROM Directory Structure

NVDM2	NVDM/2 subdirectory
EXE	
CM2111IN.PRO	
CMDLINE.PRO	
CONNECT.PRO	
DB2CE211.PRO	
DB2CS12.PRO	
DB2SU211.PRO	
DB2SUSWI.PRO	
DB2SV211.PRO	
DISKPREP.PRO	
DSPINSTL.PRO	
IP07045.PRO	
LAPSINST.PRO	
LS301REQ.PRO	
LS301SRV.PRO	
LS40REQ.PRO	
LS40SRV.PRO	
LS50REQ.PRO	
LS50SRV.PRO	
MINSTALL.PRO	
MPTS.PRO	
NDM2INST.PRO	
NVDM21.RSP	
NVDM2SRV.RSP	
NVDMCLI.PRO	
NVDMUPD.PRO	
OS211INS.PRO	
OS2V3INS.PRO	
PCOS2V41.PRO	
PRINTER.PRO	
REXXSUPT.PRO	
RMPI.PRO	
SEMNT211.PRO	
TCPIP30.PRO	
WORD.PRO	
WR08150.PRO	
WR08150.RSP	
IP08152.PRO	
IP08152.RSP	

Figure 119 (Part 2 of 8). The Sample Code CDROM Directory Structure

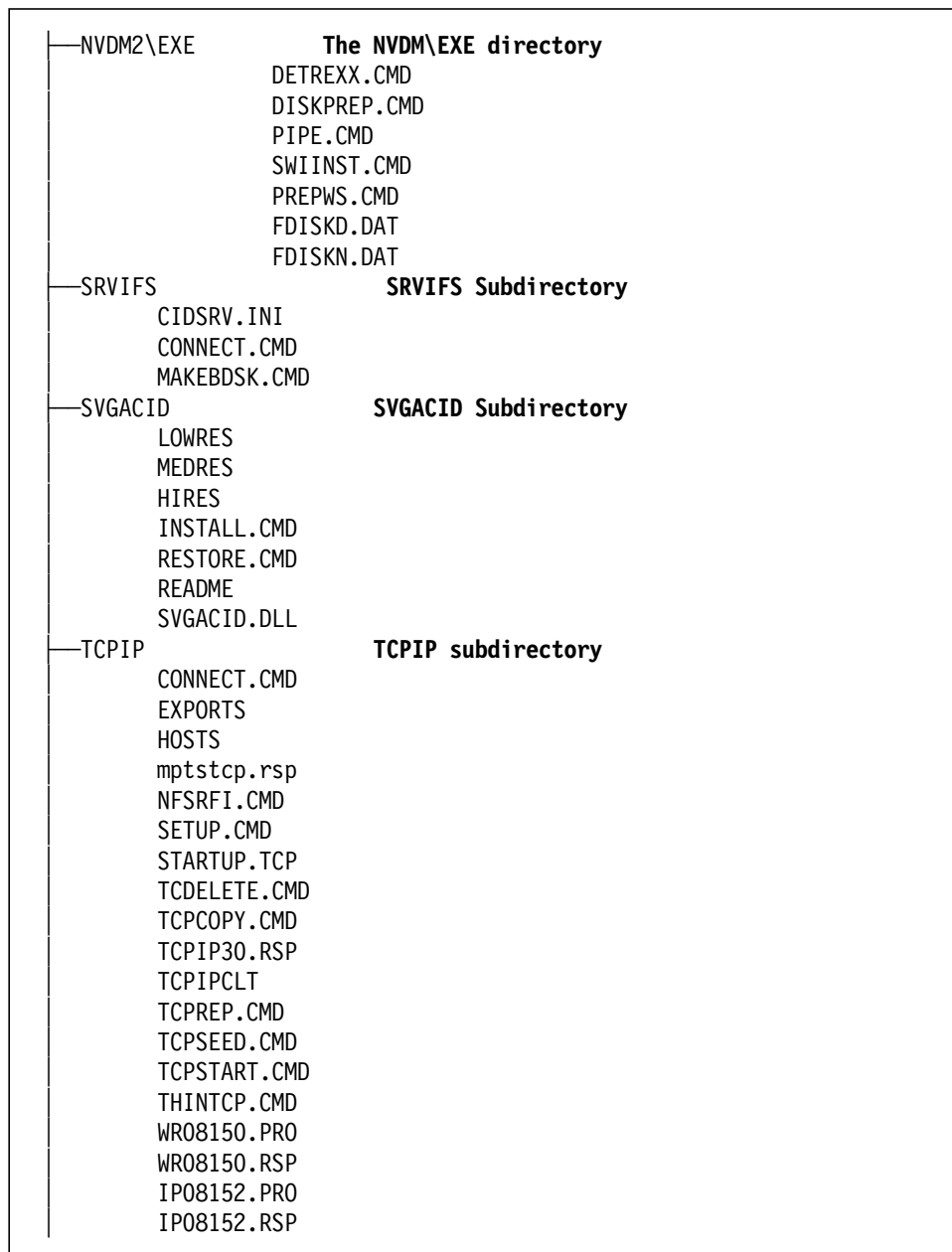


Figure 119 (Part 3 of 8). The Sample Code CDROM Directory Structure

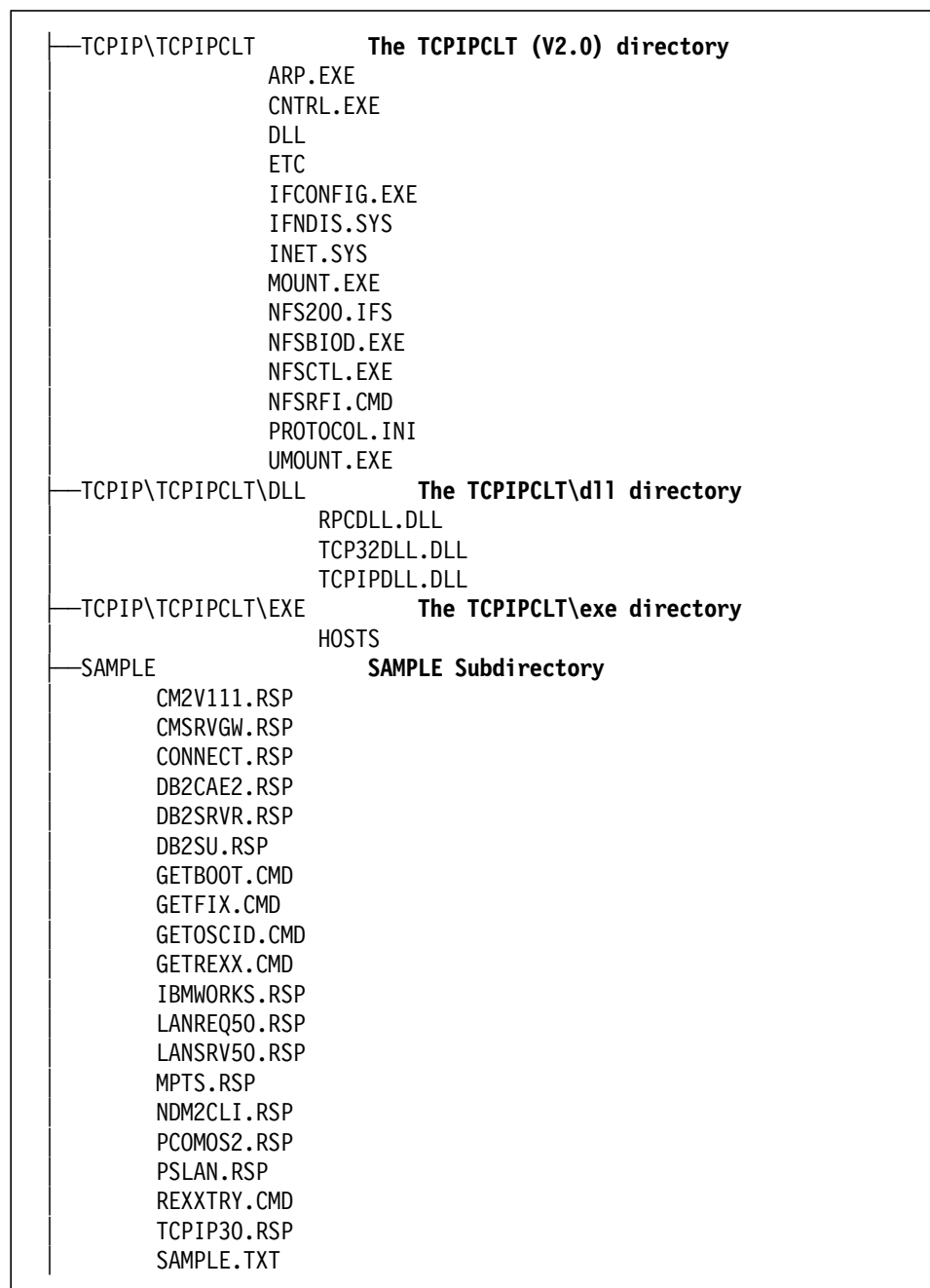


Figure 119 (Part 4 of 8). The Sample Code CDROM Directory Structure

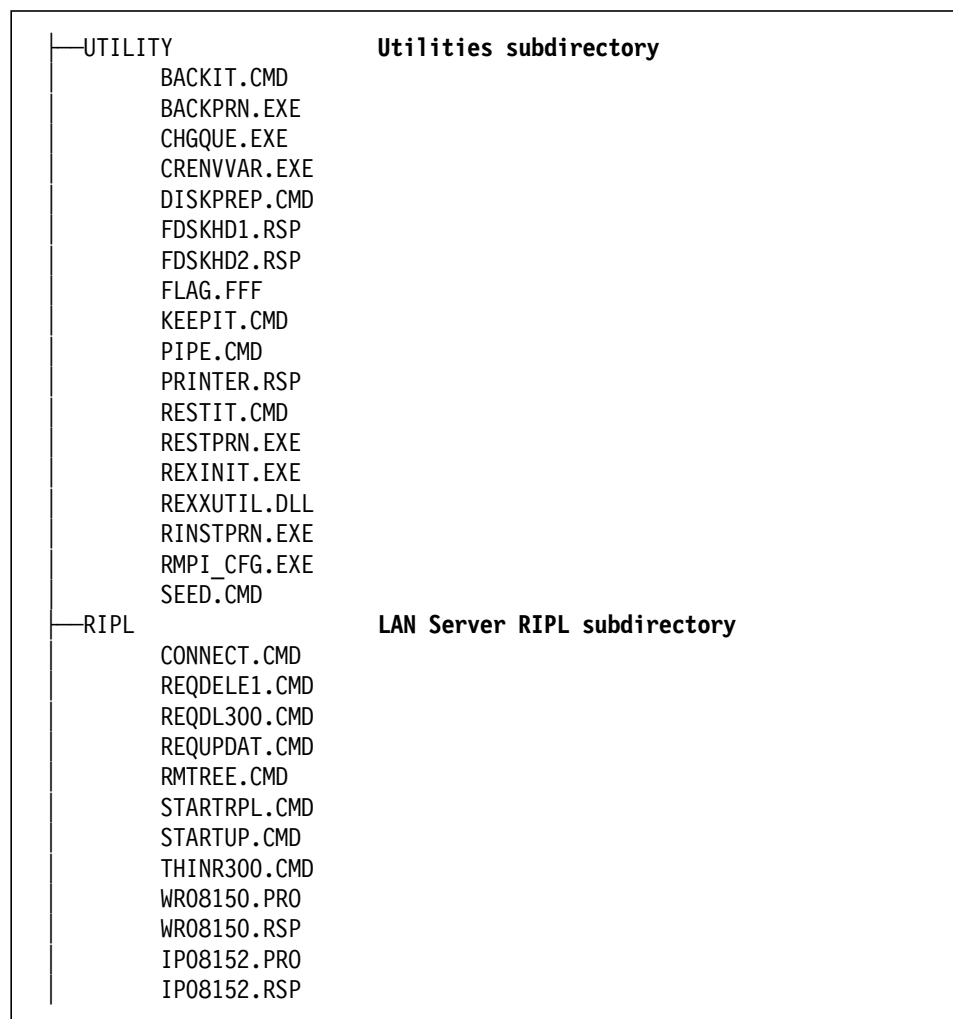


Figure 119 (Part 5 of 8). The Sample Code CDROM Directory Structure

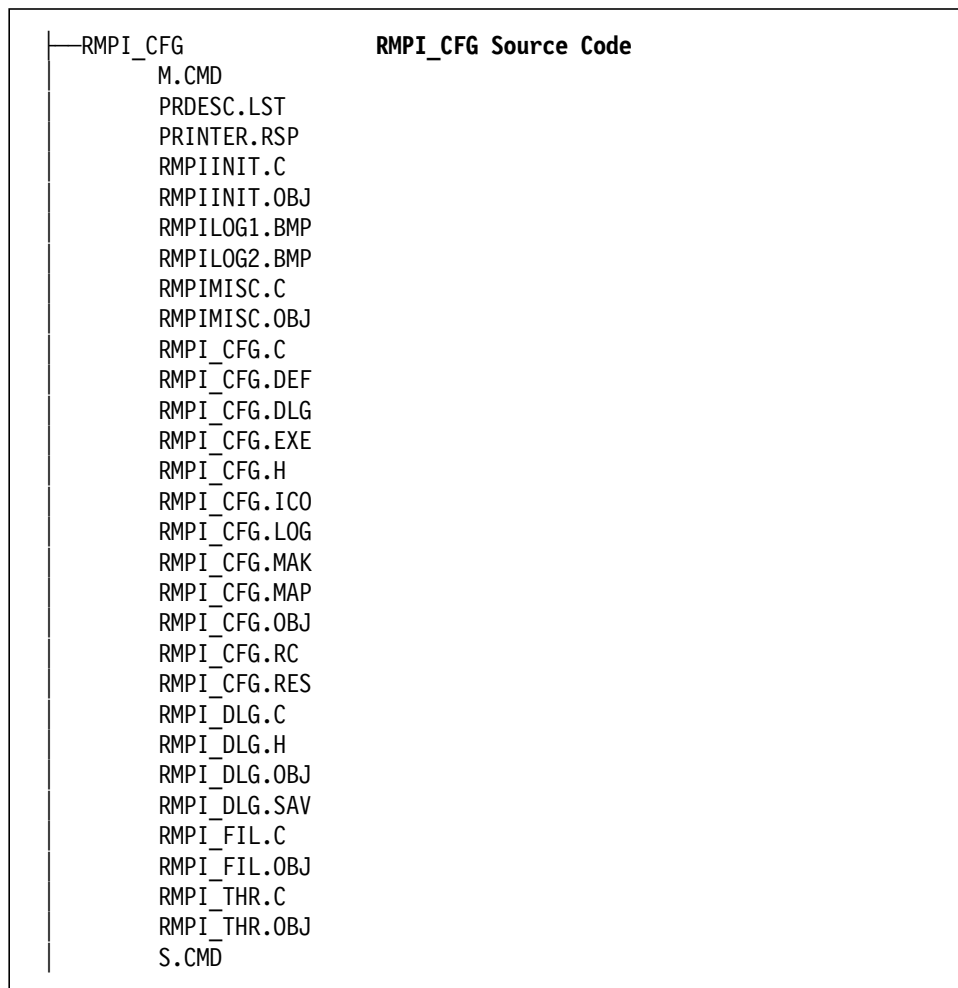
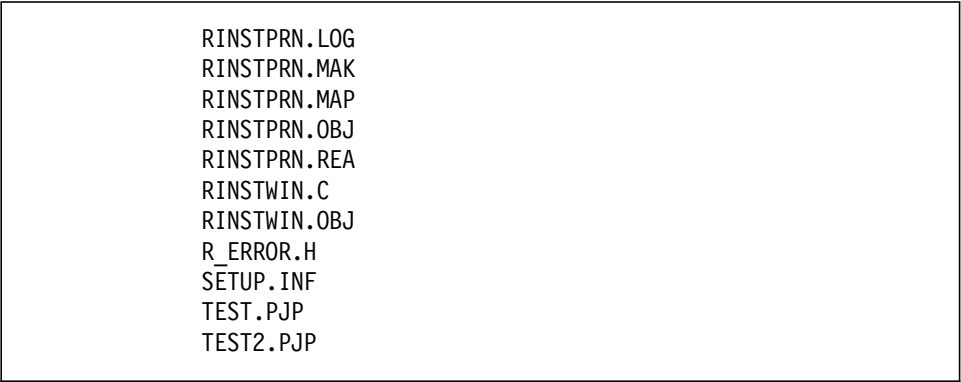


Figure 119 (Part 6 of 8). The Sample Code CDROM Directory Structure

└─RINSTPRN	RINSTPRN Source Code
	APPS.INF
	BACKPRN.C
	BACKPRN.DEF
	BACKPRN.EXE
	BACKPRN.MAP
	BACKPRN.OBJ
	CHGQUE.C
	CHGQUE.DEF
	CHGQUE.EXE
	CHGQUE.MAP
	CHGQUE.OBJ
	CONTROL.INF
	DRVMAP.INF
	LASER2.PJP
	LOG.C
	LOG.OBJ
	M.CMD
	OBJ.H
	PRINTER.RSP
	RESPCHK.C
	RESPCHK.OBJ
	RESPONSE.C
	RESPONSE.OBJ
	RESTPRN.C
	RESTPRN.DEF
	RESTPRN.EXE
	RESTPRN.MAP
	RESTPRN.OBJ
	RINSTDRV.ASM
	RINSTDRV.C
	RINSTDRV.OBJ
	RINSTMSC.C
	RINSTMSC.OBJ
	RINSTPJP.C
	RINSTPJP.OBJ
	RINSTPRN.C
	RINSTPRN.DEF
	RINSTPRN.EXE
	RINSTPRN.H

Figure 119 (Part 7 of 8). The Sample Code CDROM Directory Structure



RINSTPRN.LOG
RINSTPRN.MAK
RINSTPRN.MAP
RINSTPRN.OBJ
RINSTPRN.REA
RINSTWIN.C
RINSTWIN.OBJ
R_ERROR.H
SETUP.INF
TEST.PJP
TEST2.PJP

Figure 119 (Part 8 of 8). The Sample Code CDROM Directory Structure

Appendix I. Hardware and Software Dependencies

This appendix describes some hardware and software dependencies the administrator should be aware of in order to successfully install OS/2.

I.1 OS/2 Versions and CID

The table below shows the level of OS/2 that can be remotely installed using CID techniques:

<i>Table 22. OS/2 Versions and CID. Summary of OS/2 versions that can be remotely installed using CID.</i>		
OS/2 Version	Syslevel	CID
OS/2 V2.0	6000	YES
OS/2 V2.0 Service Pak	6055	YES
OS/2 V2.00.1 (USA) Preload	6005	NOT Supported
OS/2 V2.01 (EMEA) Preload	6005	NOT Supported
OS/2 V2.0 MultiMedia	6005	Response File Installation from CDROM
OS/2 V2.1	2010	YES
OS/2 V2.1 Service Pak	6200	YES
OS/2 V2.11 (updated 2.1 Version including Service Pak)	6200	YES
OS/2 Warp V3	3000	YES
OS/2 Warp with WinOS2 V3		YES

I.1.1 OS/2 Warp V3 and OS/2 Warp with WinOS2 V3

I.1.1.1 OS/2 Warp V3

If Windows 3.1 applications should be run under OS/2 Warp V3, DOS and Windows must be installed on the system before the installation of OS/2 Warp V3.

The following will be valid installations for OS/2 Warp V3:

- Installation on a new machine with no operating system installed
Windows applications will not be supported in this case under OS/2 Warp V3. DOS and OS/2 applications are supported.
- Installation on top of system with DOS 3.3 or later

Windows applications will not be supported in this case under OS/2 Warp V3. DOS and OS/2 applications are supported.

- Installation on top of a system with DOS 3.3 or later and
 - Windows 3.1 or
 - Windows 3.11 or
 - Windows for Workgroups 3.1 (the networking function is not supported) or
 - Windows for Workgroups 3.11 (the networking function is not supported)
- IBM OS/2 V2.1 for Windows V3.1
- IBM OS/2 V2.1 for Windows V3.1 with Service Pak XR06300

Which would be the same as OS/2 V2.11 for Windows V3.1

If OS/2 Warp V3 is installed on a system with OS/2 V2.x installed the OS/2 boot drive should be formatted first!

I.1.1.2 OS/2 Warp with WinOS2 V3

Since OS/2 Warp with WinOS2 V3 includes everything to run DOS, Windows 3.1 and OS/2 applications there is no prerequisite.

The following will be valid installations for OS/2 Warp V3:

- Installation on a new machine with no operating system installed
- Installation on top of system with DOS 3.3 or later
- Installation on top of a system with DOS 3.3 or later and
 - Windows 3.1 or
 - Windows 3.11 or
 - Windows for Workgroups 3.1 (the networking function is not supported) or
 - Windows for Workgroups 3.11 (the networking function is not supported)
- Installation on top OS/2 V2.0

This is not verified at the time of writing, May 1996.

- Installation on top OS/2 V2.1
- Installation on top OS/2 V2.11

At the time of writing this book we do not know if OS/2 Warp with WinOS2 V3 can be installed on top of IBM OS/2 V2.1 for Windows V3.1 or not.

I.2 Loadable BIOS and CID

There are three different scenarios for the installation of loadable BIOS files depending on the machine type:

- **Systems with resident BIOS** indicates the BIOS code is in ROM and that no loadable BIOS files are required.
- **Systems requiring loadable BIOS** indicates that at least one loadable BIOS file is required (resident BIOS may also be present).
- **Systems without System Partition** indicates the system has no system partition installed on fixed disk.

The following figure summarizes the different installation scenarios of the loadable BIOS files:

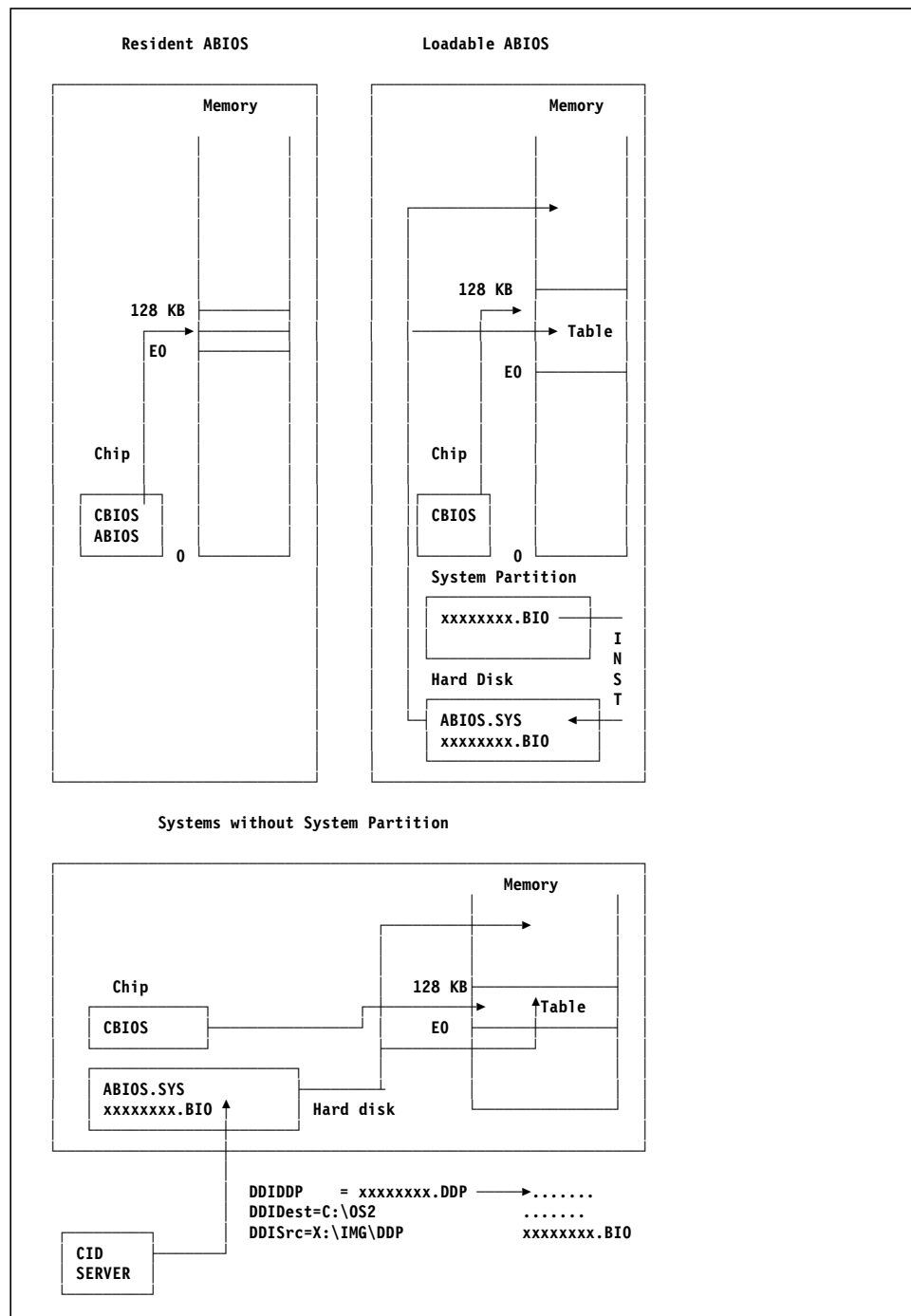


Figure 120. Loadable ABIOS Installation Scenarios

Systems having a system partition will load the loadable BIOS file from the system partition during OS/2 V2.1 installation.

Systems without a system partition will load the loadable BIOS file from their reference diskette during standard diskette installation or load the loadable BIOS files from the code server according to the DDIDDP, DDIDest and DDISrc keywords in the response file. See Appendix C, "OS/2 Response File Keywords" on page 433 for the usage of these keywords.

Note: OS/2 V2.1 will prompt the user for the reference diskette during normal OS/2 V2.1 diskette installation.

The following table summarizes the type of machines having resident BIOS, system partition or loadable BIOS. The last column shows if any particular action has to be taken during a remote CID installation of OS/2 V2.1.

<i>Table 23. OS/2 V2.0 and Machine Types. Summary of machines by BIOS type.</i>				
Machine Type	Resident BIOS	System Partition	Loadable BIOS	CID Action
Model 35/40 8525, 2123, VP	NO	NO	NO	NO
PS/2	YES	YES/NO (depending on model)	NO	NO
PS/2 9556/57, 9585/95	NO	YES	YES	NO
PC Series 300	NO	YES	YES	NO
PC Series 500	NO	YES/NO (depending on model)	YES	YES/NO (depending on system partition)
PC Series 700	NO	YES	YES	NO
ThinkPad 700, 720 series	NO	NO	YES	YES
ThinkPad 750, 350, 500 series	NO	NO	NO	NO

The ThinkPad 700/720 series is the only machine type where the administrator should specify the DDIDDP, DDIDest and DDISrc keywords in the OS/2 V2.x response file in order to install the loadable BIOS files. See Appendix C, "OS/2 Response File Keywords" on page 433 for a description of these keywords.

Additionally, the boot diskettes have to be updated with the BIOS files:

- Copy the .BIO file from the system reference diskette to the OS/2 V2.x DISK 1 diskette.
- Edit the BIOS.SYS on the DISK1 diskette and insert the name of the .BIO as the first line of the BIOS.SYS file.

For more information on ThinkPad systems, please refer to *ThinkPad Systems*, GG24-4297.

I.3 PCMCIA and CID

PCMCIA

- stands for Personal Computer Memory Card International Association
- is a bus architecture that is mainly implemented in laptop computers like the IBM Thinkpad series
- adapters are credit card size and are also available in several types (called "Type I", "Type II" and "Type III") that are different in the unit's height
- cards are available in a rich variety today : LAN adapters (like Token Ring and Ethernet), Host adapters (like 3270-Emulation), modems, ATA-cards (small hard disk drives), memory ... and many more.

As far as OS/2 Warp V3 is concerned, PCMCIA support consists mainly of three layers of software:

- PCMCIA Card Services
 - statement in CONFIG.SYS: BASEDEV=PCMCIA.SYS
 - this entry must precede the entry for PCMCIA Socket Services
- PCMCIA Socket Services
 - are hardware dependent
 - statement in CONFIG.SYS: BASEDEV=IBM2SS01.SYS
 - this entry must follow the entry for PCMCIA Card Services
 - this in an example valid only for IBM Thinkpad 750
 - if you install additional products (after the installation of Socket Services) that modify the CONFIG.SYS file by adding device drivers, you may experience problems with some PCMCIA cards. If this happens to you, try to resolve your problems by moving all PCMCIA related drivers to the bottom of your CONFIG.SYS file.
- PCMCIA Super Client Drivers (may appear in any order)
 - Modem driver
 - ATA (hard disk) driver
 - FLASH memory driver

Note that there are major differences between OS/2 Warp V3 and previous OS/2 releases such as OS/2 V2.11. To mention a few of them:

- The resource map utility (ICRMU01.SYS) is no longer needed. In OS/2 V2.11 you had to copy it from the Thinkpad Utility Diskette to hard disk and add a DEVICE statement to CONFIG.SYS. OS/2 Warp V3 has this function built in.
- OS/2 V2.11 installation included only Card Services but no Socket Services. OS/2 Warp V3 covers both.
- The CID utilities SEDISK and SEMAINT offer a new parameter /P to provide for and select the appropriate PCMCIA support.

Additional information about OS/2 Warp V3, Thinkpads, PCMCIA, Card and Socket Services can be found in *OS/2 Warp Version 3 and BonusPak "Exploring a New Generation"* GG24-4426 and *ThinkPad Systems* GG24-4297

Installing systems with PCMCIA

If you are installing systems with PCMCIA LAN adapter cards, perform the following steps: These have been tested in our lab using an IBM ThinkPad Model 750 equipped with an IBM Token-Ring 16/4 Autoringspeed Credit Card Adapter.

Creating Client Boot Diskettes

1. Determine the PCMCIA support that you need:
 - Use an editor like EPM to look at the file PCMCIA.TBL that can be found in the directory OS2INSTALL. The information in this file looks similar to the PCMCIA section in the OS/2 sample response file SAMPLE.RSP.
 - Note the number of the line that describes your hardware. In our lab example this number was: 17 = IBM ThinkPad 750. This number will be passed to the /P: parameter of SEDISK in the next step.
2. Run SEDISK to create the two boot diskettes.
 - A detailed description of SEDISK can be found in 15.1.2, "SEDISK" on page 377 or in the file README.CID in directory F:SHARE_AIMGCONNECTDISK_0.
 - Provide the line number from the previous step as parameter /P:<line number>. This directs SEDISK to include PCMCIA support on the second boot diskette.
 - Insert the two diskettes as prompted by SEDISK. Leave the second diskette in the drive for the following steps.

— SEDISK example as used in our lab (enter as a single line) —

```
F:\SHARE_A\EXE\CONNECT\SEDISK  
/S:F:\SHARE_A\IMG\CONNECT  
/T:A:  
/P:17
```

3. Use THINLAPS to add LAN transport to the second boot diskette.

- Determine the name of the NIF file that supports the LAN adapter used for the installation. Information about supported adapters can be found in IBMCOMMACSREADMAC.TXT. In our lab we used IBMTOKCS.NIF for the IBM Token-Ring 16/4 Autoringspeed Credit Card Adapter.
- If your adapter is not listed as supported by MPTS, you need to update the code servers MPTS image files. See E.3.1, “Support for Additional Drivers” on page 534 and/or IBMCOMREADME.MTP for information on this task.

— THINLAPS example as used in our lab (enter as a single line) —

```
F:\SHARE_A\IMG\MPTS\THINLAPS  
F:\SHARE_A\IMG\MPTS  
A:\  
IBMTOKCS.NIF
```

- After completion of THINLAPS edit the file A:PROTOCOL.INI
 - Check parameters RINGSPEED and AUTORINGSPEED. Use only one of them!
 - In our lab we used AUTORINGSPEED.
 - If you use RINGSPEED instead, check whether it is set to the desired value (4 or 16 Mbps).
 - Only NetBIOS should be specified as networking protocol.

4. Add the appropriate client support to the second boot diskette

- For NetView DM see 14.5.1, “Creation of Boot Diskettes” on page 356
- For LCU see 11.5, “Preparation of Client Workstations” on page 302

5. Remove the second boot diskette from the drive. The boot diskettes are now ready for use.

Preparing Response Files for OS/2

The OS/2 Warp V3 installation program handles the installation of PCMCIA Card and Socket Services as well as Super Client drivers. In addition to the response file parameters already discussed in Appendix C, “OS/2 Response File Keywords” on page 433, you should set:

- APM=2 (Install)
- PCMCIA=<number of your hardware> (in our lab: 17 = Thinkpad 750)
- PCMCIAOptions=2 or
PCMCIAOptions=3 (as recommended in the next paragraph)

Known PCMCIA related problems

The installation programs of IBM LAN Server/Requester (both Version 4 or 5) have a problem, if they are executed on a system with WARP’s PCMCIA FLASH device drivers installed:

- The installation starts and finishes within a few seconds.
- No error message appears or is logged.
- Instead, if you are doing a CID installation, you will get a feedback like “Installation successful”, although no LAN software has been installed.

There are basically two ways to deal with this situation:

1. Do not install the PCMCIA FLASH device drivers. In the OS/2 response file ...
 - do not specify one of these

<i>Table 24. Results of PCMCIAOptions settings in OS/2 response file. Settings that do affect LAN Server/Requester installation</i>	
Parameter	Installs...
PCMCIAOptions=1	All services
PCMCIAOptions=4	FLASH services

- you can safely specify one of these

<i>Table 25. Results of PCMCIAOptions settings in OS/2 response file. Settings that do not affect LAN Server/Requester installation</i>	
Parameter	Installs...
PCMCIAOptions=2	Modem/FAX services
PCMCIAOptions=3	Hard disk services

- if you need the combination of the "safe" options 2 and 3, you cannot install both of them in a one-step installation, because

<i>Table 26. Results of PCMCIAOptions settings in OS/2 response file. Possible combinations of "safe" PCMCIAOptions</i>	
Parameter(s)	Result
PCMCIAOptions=2,3	Invalid: Response file syntax error.
PCMCIAOptions=2 3	Invalid: Will install no PCMCIA support at all.
PCMCIAOptions=2 PCMCIAOptions=3	The combination of two lines containing a PCMCIAOption results in the installation of the latter option, as the last occurrence of this response file parameter overwrites all of its predecessors.

If you need to install more than one PCMCIAOption, you might choose one of these ways:

- Use a separate change file to copy the required drivers and add the related entries to CONFIG.SYS. You can find information about this by searching the OS/2 command reference for "PCMCIA", which is also available online.
 - Install one PCMCIAOption during the first OS/2 installation and another PCMCIAOption as an update.
 - Install all drivers by specifying PCMCIAOptions=1 and proceed as described in the next paragraph.
2. If you already have the PCMCIA FLASH device drivers installed on your system, you can edit your CONFIG.SYS and comment these lines out:

```
DEVICE=<drive>:OS2BOOTICMEMMTD.SYS
DEVICE=<drive>:\OS2\BOOT\ICMEMCDD.SYS
```

After rebooting the system with the modified CONFIG.SYS the installation programs of IBM LAN Server/Requester can be successfully executed.

Remark: We did not spend extensive time on tests with IBM LAN Server/Requester and FLASH services. However, it might be useful to mention that we had no problems logging on to a LAN server, if we re-activated the FLASH drivers again after a successful installation of the LAN requester.

I.4 RAID and CID

If you are installing a system with a RAID controller, there are additional tasks to be done to get the system to work.

Before you can start the automated install, you have to configure the RAID system. This has to be done with the system booted from the RAID diskette. Then, the disks have to be initialized and synchronized. Please follow the manuals that come with the system for these tasks. There is no way to execute these tasks remotely in a CID scenario. Do not forget to partition the system as usual before installing the operating system.

The RAID controller needs a specific device driver that is not part of OS/2 but is delivered on the RAID diskette that comes with the system. This device driver has to be on the boot diskettes and integrated in the OS/2 install. If this is not done, the system will show inconsistencies in the handling of the hard drives. Therefore, the following changes have to be made in the CID scenario:

Creating Client Boot Diskettes

Run SEDISK to create the two boot diskettes. See 15.1.2, “SEDISK” on page 377. The CONFIG.SYS of the boot diskette has to be updated with the device driver for the RAID controller. Copy the driver on the diskette and add the following line to the CONFIG.SYS on the diskette, before all other BASEDEV commands:

```
BASEDEV=IBMRAID.ADD
```

Continue creating the boot diskettes normally.

Editing the OS/2 Response File

To install the device driver for the RAID controller during operating system install, the DDI keywords can be used. Please see Appendix C, “OS/2 Response File Keywords” on page 433 for a description of these keywords. Create a directory on the code server to hold the files needed for the RAID controller, and copy those files from the RAID diskettes to this subdirectory. Edit the operating system response file as follows, assuming that a directory IMGRAIDV20 was created and the operating system install is done to drive C::

```
DDISrc  = X:\IMG\RAIDV20
DDIDest = C:\
DDIDDP  = IBMRAID.DDP
```

The icon for the RAID controller utility is not created during this install. This can be done during the follow-on install of the system.

Appendix J. CID Enabled Applications

The following list gives an overview of IBM and Independent Software Vendor (ISV) applications which are CID enabled. The list is dated as of January, 1996. Please be aware that not all products may be available in all countries.

J.1 IBM Products

<i>Table 27 (Page 1 of 6). CID Enabled IBM Products</i>	
Product	Availability
3130 Advanced Function Printer Model 03S	96Q1
A Programming Language 2/2 V1.0 (APL2/2)	NOW
ADSTAR Distributed Storage Manager/2 V1.2 (ADSTAR DSM/2)	NOW
ADSTAR Distributed Storage Manager/400 V1.2 for OS/400 V2.3	NOW
Advanced Peer-to-Peer Networking Topology and Accounting Management (APPNTAM) Feature - NetView V2R2	NOW
AntiVirus Desktop Edition V2.X for DOS, Windows, and OS/2	96Q3
AntiVirus Enterprise Edition V2.X	96Q3
AnyNet APPC over TCP/IP for Windows V1.0	NOW
AnyNet IPX over SNA Gateway V1.0 for OS/2	NOW
AnyNet SNA over TCP/IP Gateway V1.0 for OS/2	NOW
AnyNet/2 V2.0	NOW
AnyNet/2 NetBEUI over SNA V1.0	NOW
AnyNet/2 Sockets over SNA Gateway V1.1	NOW
BookManager BUILD V2.0 for OS/2	NOW
BookManager READ V2.0 for Windows	NOW
COBOL Family - COBOL for OS/2	NOW
CommonPoint Application Developer Toolkit V1.0 for OS/2 Warp	NOW
CommonPoint Application System V1.0 for OS/2 Warp	NOW
Communications Manager/2 V1.0 (CM/2 V1.0)	NOW

<i>Table 27 (Page 2 of 6). CID Enabled IBM Products</i>	
Product	Availability
Communications Manager/2 V1.10 (CM/2 V1.10)	NOW
Communications Manager/2 V1.11 (CM/2 V1.11)	NOW
CoOperative Development Environment/370 V1.1	NOW
Customer Information Control System (CICS) Clients - CICS Client V1.0 for OS/2	NOW
Customer Information Control System (CICS) Clients - CICS Client V1.0 for Windows	NOW
Customer Information Control System V2.0 for OS/2 (CICS OS/2)	NOW
Customer Information Control System V2.0.1 for OS/2 (CICS OS/2)	NOW
DATABASE 2 Client Application Enabler/2 V1.2	NOW
DATABASE 2 OS/2 V1.0 (DB2/2 V1.0)	NOW
DATABASE 2 OS/2 V1.2 (DB2/2 V1.2)	NOW
DATABASE 2 World-Wide-Web (WWW) Connection Version for OS/2	NOW
DataGuide V1.1 (Administrator for OS/2)	96Q1
DataGuide V1.1 (User for OS/2)	96Q1
DataGuide V1.1 (User for Windows)	96Q1
DataRefresher V1.0	NOW
Distributed Access Control Facility V1.3	NOW
Distributed Computing Environment Runtime Client V1.0 for OS/2 (DCE Runtime Client for OS/2)	NOW
Distributed Computing Environment Software Developer's Kit V1.0 for OS/2 and Windows (OS/2 environment only)	NOW
Distributed Database Connection Services/2 V1.0 (DDCS/2 V1.0)	NOW
Distributed Database Connection Services/2 V2.2 (DDCS/2 V2.2)	NOW
Electronic Publishing Edition V2.0 for OS/2	NOW
Electronic Publishing Edition V2.0 for OS/2 (Internet Delivery)	NOW
Encina Client V1.3 for OS/2	NOW
Extended Services V1.0 for OS/2 (ES V1.0)	NOW

<i>Table 27 (Page 3 of 6). CID Enabled IBM Products</i>	
Product	Availability
Extended Services with Database Server V1.0 for OS/2	NOW
FlowMark V1.0 for OS/2	NOW
FlowMark V2.0 for OS/2	NOW
FlowMark Runtime Client for Windows	NOW
IBM Workgroup V1.0	NOW
ImagePlus VisualInfo Client for OS/2	NOW
ImagePlus VisualInfo Library Server for OS/2	NOW
ImagePlus VisualInfo Object Server for OS/2	NOW
Internet Connection Secure Server V1.1 for AIX and OS/2 Warp	NOW
Internet Connection Secure Web Explorer V1.1 for OS/2 Warp	NOW
LAN Automated Distribution/2 V3.0 (LAD/2 V3.0)	NOW
LAN Distance Connection Server V1.0	NOW
LAN Distance Connection Server V1.1	NOW
LAN Distance Remote V1.0	NOW
LAN Distance Remote V1.1	NOW
LAN Distributed Platform/2 V2.0 (LANDP/2 V2.0)	NOW
LAN NetView Agents Extended V1.0	NOW
LAN NetView Agents for DOS V1.0	NOW
LAN NetView Enabler V1.0	NOW
LAN NetView Fix V1.0	NOW
LAN NetView Manage V1.0	NOW
LAN NetView Management Utilities for OS/2 V2.0	NOW
LAN NetView Monitor V1.0	NOW
LAN NetView Start V1.1 (Start V1.1)	NOW
LAN NetView Tie V1.0	NOW
LAN Network Manager Entry V1.0 (LNME V1.0)	NOW
LAN Network Manager V1.0 (LNM V1.0)	NOW
LAN Network Manager V1.1 (LNM V1.1)	NOW
LAN Network Manager V2.0 for OS/2	NOW

<i>Table 27 (Page 4 of 6). CID Enabled IBM Products</i>	
Product	Availability
LAN Station Manager V1.0 (LSM V1.0)	NOW
MQSeries V2.0 for OS/2	NOW
MQSeries Three Tier V1.0 for OS/2	NOW
NetFinity Manager V2.0	NOW
NetFinity Manager V2.01	NOW
NetFinity Manager V3.0	NOW
NetFinity Services V2.0	NOW
NetFinity Services V2.01	NOW
NetFinity Services V3.0	NOW
NetView Distribution Management Agent/2 V1.0 (NetView DMA/2)	NOW
NetView Distribution Management Agent/DOS (NetView DMA/DOS)	NOW
NetView Distribution Manager Easy Preparer for OS/2	NOW
NetView Distribution Manager/2 V2.0 (NetView DM/2 V2.0)	NOW
NetView Distribution Manager/2 V2.1 (NetView DM/2 V2.1)	NOW
NetView V2.0 for OS/2	NOW
NetView V2.1 for OS/2	NOW
NetView Network Planner/2	NOW
NetWork Door/2 V1.0 (DOS/Windows Client Support)	NOW
NetWork Door/2 (English Version)	NOW
Network Security Program for Multiple Operating Environments V1.2 (NetSP V1.2)	NOW
Network Transport Services/2 (NTS/2)	NOW
Neural Network Utility Entry V3.1 for OS/2 (NNU Entry)	NOW
Neural Network Utility Entry V3.1 for Windows (NNU Entry)	NOW
Neural Network Utility V3.1 for OS/2 (NNU)	NOW
Neural Network Utility V3.1 for Windows (NNU)	NOW
Operating System/2 V2.0 (OS/2 V2.0)	NOW
Operating System/2 V2.1 (OS/2 V2.1)	NOW
Operating System/2 V2.1 Special Edition for use with Windows Version 3.1 (OS/2 for Windows)	NOW

<i>Table 27 (Page 5 of 6). CID Enabled IBM Products</i>	
Product	Availability
Operating System/2 V2.11 for Symmetrical Multiprocessing (OS/2 for SMP)	NOW
Operating System/2 Warp V3.0 (OS/2 Warp)	NOW
Operating System/2 Warp V3.0 with WINOS	NOW
OS/2 LAN Server - Advanced V3.0	NOW
OS/2 LAN Server - Advanced V4.0	NOW
OS/2 LAN Server - Entry V3.0	NOW
OS/2 LAN Server - Entry V4.0	NOW
OS/2 LAN Server for Macintosh V1.0	NOW
Personal Application System Builder/2 V3.0	NOW
Personal Application System/2 V3.0	NOW
Personal Communications AS/400 V4.0 for OS/2	NOW
Personal Communications/3270 V4.0 for OS/2	NOW
Personal Computer DOS V6.1 (PC DOS V6.1)	NOW
Personal Computer DOS V6.3 (PC DOS V6.3)	NOW
Personal Computer DOS V7.0 (PC DOS V7.0)	NOW
Point-Of-Sale Subsystem V1.0 for OS/2 (POSS V1.0)	NOW
Presentation Manager Office/2 V1.3 (PMO/2 V1.3)	NOW
Presentation Manager Office/2 V3.0 (PMO/2 V3.0)	NOW
SAA Consumer Transaction Definition/2 V2.1	NOW
SAA Consumer Transaction Runtime/2 V2.1	NOW
SOMobjects Developer Toolkit V1.0 for OS/400	NOW
SmallTalk V3.0 for OS/2	NOW
Software Installer V1.2 for OS/2 (SI for OS/2)	NOW
StorePlace Application Function Library for OS/2	NOW
StorePlace Customer Notebook for OS/2	NOW
StorePlace Distributed Data Services for OS/2	NOW
StorePlace Sales Analyst for OS/2	NOW
StorePlace Time and Attendance for OS/2	NOW
StorePlace Workbench for OS/2	NOW
StorePlace Workforce Planner for OS/2	NOW

<i>Table 27 (Page 6 of 6). CID Enabled IBM Products</i>	
Product	Availability
System Performance Monitor/2 V2.0 (SPM/2 V2.0)	NOW
SystemView for OS/2	NOW
SystemView License Management Application Developer's Toolkit V1.0 for OS/2	NOW
SystemView License Management Runtime V1.0 for OS/2	NOW
TeamConnection V1.0 for OS/2	NOW
Transmission Control Protocol/Internet Protocol V2.0 for OS/2 (TCP/IP V2.0 for OS/2)	NOW
Turboways 25 ATM Microchannel Adapter	96Q1
VisualAge C++ V3.0 for OS/2	NOW
VisualAge C++ V3.0 for OS/2 Open Class Library Source	NOW
VisualAge for SmallTalk V3.0 for OS/2 and Windows	NOW
Visualizer Family V1.2	NOW
Visualizer Charts for OS/2	NOW
Visualizer Development for OS/2	NOW
Visualizer Plans for OS/2	NOW
Visualizer Procedures for OS/2	NOW
Visualizer Query for AIX/6000	NOW
Visualizer Query V1.0 for OS/2	NOW
Visualizer Query V1.1 for OS/2	NOW
Visualizer Query V1.2 for OS/2	NOW
Visualizer Query V1.0 for Windows	NOW
Visualizer Query V1.0 with DB2/2 V1.2 (Single User)	NOW
Visualizer Query V1.1 with DB2/2 V1.2 (Single User)	NOW
Visualizer Statistics for OS/2	NOW
Visualizer UltiMedia Query for OS/2	NOW
Workstation Interactive Test Tool V1.1 for Windows (WITT for Windows)	NOW

J.2 Independent Software Vendor Products

The following list of CID enabled applications was compiled by IBM from information supplied by independent software vendors. Each vendor has represented that the products listed are enabled according to the CID Enablement Guidelines. IBM does not verify that the vendors have actually shipped the application, and does not test to ensure that the application is CID enabled.

<i>Table 28 (Page 1 of 5). CID Enabled Products by Other Vendors</i>	
Company, City, US State	
Product	Availability planned
Abacus, Grand Rapids, MI	
OS/2 2.1 Bible	NOW
Abraxas Software, Portland, OR	
CodeCheck	NOW
Pcyacc	NOW
AutoSoft Inc., Roswell, GA	
MainScript	NOW
Automation Consultants International, Mission Viejo, CA	
CATALIST/PC	NOW
BGS Systems, Inc., Waltham, MA	
Analyze for OS/2	NOW
Baron Software Services, Inc., Massapequa Park, NY	
Manage - iT	NOW
Canyon Software Corporation, New York, NY	
JCL Navigator	NOW
Capstone Software, Inc., Carmel, IN	
SpaceMap 1.1	NOW
Chipchat- Cawthon Software, Dearborn, MI	
ChipChat Communications Objects	NOW
Citation Software, Inc., Nashua, NH	
Reply Mail Designer	NOW
Commix SP, Inc., McLean, VA	
DisplayMaster for OS/2	NOW

<i>Table 28 (Page 2 of 5). CID Enabled Products by Other Vendors</i>	
Company, City, US State	
Product	Availability planned
Computer Systems Integration, Inc., Providence, RI	
FaxForward (tm)	NOW
Compuware Corporation, Farmington Hills, MI	
Remote Control/II	NOW
Creative Systems Programming Corp., Mt. Laurel, NJ	
Golden CommPass	NOW
Crossen Computing, Vienna, VA	
Johnny AppleRead	NOW
Dynamic Object Language Group, Haverhill, MA	
Yolambda	NOW
FaxPro Corporation, Solana Beach, CA	
FAXPRO 4 Database	NOW
FAXPRO 4 OS/2	NOW
FAXPRO 4 Routing	NOW
FAXPRO 4 Translation	NOW
Footprint Software Inc., Toronto, Canada	
Footprint Works For OS/2	NOW
GRAFTech Development Corporation, Westland, MI	
BARCODES-PLUS	NOW
Hilbert Computing, Olathe, KS	
Chron	NOW
IRMWare Services, Phoenix,AZ	
Employee Development Management System	NOW
Entity/Relationship Diagrammer	NOW
ImageSoft, Inc., Port Washington, NY	
AM/ST	NOW
CommonBase	NOW
CommonView	NOW
Glockenspiel C + +	NOW
ImagingObjects	NOW

<i>Table 28 (Page 3 of 5). CID Enabled Products by Other Vendors</i>	
Company, City, US State	
Product	Availability planned
Object/Designer	NOW
Object/Developer	NOW
Tools++	NOW
VZ Programmer	NOW
Intec Controls Corporation, Walpole, MA	
Paragon TNT	NOW
Integrated InfoNet Technology Inc., Irvine, CA	
BusinessLink (tm) 1.0	NOW
Intelligent Environments, Tewksbury, MA	
AM (Applications Manager)	NOW
AM - CP Workbench	NOW
AM - Hostview Workbench	NOW
AM - SQL Workbench	NOW
Intersoft Systems Inc., Norcross, GA	
CONCOURSE - TP V2.0	NOW
KnowledgeNet Inc., Palatine, IL	
Net/Wrk OS2	NOW
Lotus Development Corporation Headquarters, Cambridge, MA	
Freelance Graphics for OS/2 V2.1	NOW
Lotus 1-2-3 for OS/2 V2.1	NOW
MAXM Systems Corporation, Vienna, VA	
MAX/MAP	NOW
MAXM Operator Workstation	NOW
MEI-SHU Computer & Communications Co, Gaithersburg, MD	
VC2000	NOW
MISTIK Systems, Ann Arbor, MI	
UnTie Com	NOW
Magus, Mountain View, CA	
Magus PageTurner for OS/2	NOW
Microformatic, S. Windsor, CT	

<i>Table 28 (Page 4 of 5). CID Enabled Products by Other Vendors</i>	
Company, City, US State	
Product	Availability planned
Fax/PM LAN	NOW
Pacific Gold Coast Corporation, Glen Cove, NY	
PGC CASE Graphics V1.0	NOW
Pinnacle Technology, Inc., Kirklin, IN	
Desktop Observatory	NOW
Porak Computing Services Inc., Colorado Springs, CO	
Architectural Construction System (ArchCon)	NOW
Bill of Materials Reporting System (BCMRS)	NOW
Office Furniture System (OFS)	NOW
Proportional Software Corporation, Fort Collins, CO	
DCF/2	NOW
QMI, Annapolis, MA	
Quantitative Sentinel	NOW
Raleigh Systems, Cleveland, OH	
Caliber 32 (tm)	NOW
S-Cubed, Inc., Stamford, CT	
Developer's Assistant for Client Server Systems	NOW
Developer's Assistant for Information Systems	NOW
Secure User Programming by Refinement (SuPRe)	NOW
SMART Communications, New York, NY	
SMART Expert Editor	NOW
SMART Translator English-French	NOW
SMART Translator English-German	NOW
SMART Translator English-Italian	NOW
SMART Translator English-Spanish	NOW
SMART Translator French-English	NOW
SMART Translator German-English	NOW
SMART Translator Italian-English	NOW
SMART Translator Spanish-English	NOW
SofNet, Marietta, GA	

<i>Table 28 (Page 5 of 5). CID Enabled Products by Other Vendors</i>	
Company, City, US State	
Product	Availability planned
FaxWorks PRO LAN for OS/2	NOW
FaxWorks PRO for OS/2	NOW
Software Corporation of America, Stamford, CT	
PolyPM/2	NOW
TalkThru for OS/2 (V 2.2)	NOW
Sundial Systems Corporation, Seal Beach, CA	
Relish(r) 32-BIT	NOW
Relish(r) NET 32-BIT	NOW
Syntegration, Chino, CA	
The Secure Workplace for OS/2	NOW
Syntelligence Systems Inc., Mountain View, CA	
Lending Advisor	NOW
SynCore	NOW
Underwriting Advisor	NOW
Systems & Software, Holmdel, NJ	
SYSEN	NOW
TASCO, Inc., Foster City, CA	
PlantMaster	NOW
Tangram Enterprise Solutions, Inc., Raleigh, NC	
AM:PM Electronic Software Distribution System	NOW
TruData, Inc., Boca Raton, FL	
OS/2 Office Productivity Tool	NOW
OS/2 Retail Terminal	NOW
University Debit Card Application	NOW
Western Thunder, Sacramento, CA	
ONSPEC	NOW
WORKPLACE	NOW

Appendix K. CID Installation Messages and Return Codes

K.1 OS/2 CID Utilities' Error Messages

When using the Multiple Transport Services (MPTS) product to perform CID installations of OS/2 and other applications, error messages can be found in the MPTS *Messages and Problem Determination Guide*.

However, there are some errors that are not documented in this book; specifically, the ones that begin with the prefix CID. These error messages are associated with the four utilities that ship with OS/2 for the purpose of CID-enabling OS/2: SEDISK, SEIMAGE, SEINST, SEMAINT.

The following shows the possible error messages that may occur with each of these utilities.

```
;*****  
; Messages for 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
;
;*****
; Messages for 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
;
;*****
; Messages for SEDISK
;*****
;
CID0100E: You have entered one or more invalid arguments.
CID0106E: Failed to update %1.
CID0107E: Failed to create boot diskette.
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.

```

K.2 RSPINST Return Codes

Errors that are returned from RSPINST.EXE (the OS/2 CID installation executable) have been difficult to diagnose as they have been hard to find. This documents the 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 Connect V3.

```
*****
      The following messages are used for LAN Install
      and Response file logging, messages, errors, etc.
*****
```

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

```
*****
      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.

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.

Messages 960 - 989 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

Messages 980 - 989 are used for
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.

*****
        All other messages are error messages.
*****

INS1000: System Installation detected an internal error(00).
INS1001: System Installation detected an internal error(01).
INS1002: System Installation detected an internal error(02).
INS1003: System Installation detected an internal error(03).
INS1004: System Installation detected an internal error(04).
INS1005: System Installation detected an internal error(05).
INS1006: System Installation detected an internal error(06).
INS1007: System Installation detected an internal error(07).

```

INS1008: System Installation detected an internal error(08).
 INS1009: System Installation detected an internal error(09).
 INS1010: System Installation detected an internal error(10).
 INS1011: System Installation detected an internal error(11).
 INS1012: System Installation detected an internal error(12).
 INS1013: System Installation detected an internal error(13).
 INS1014: System Installation detected an internal error(14).
 INS1015: System Installation detected an internal error(15).
 INS1016: System Installation detected an internal error(16).
 INS1017: System Installation detected an internal error(17).
 INS1018: System Installation detected an internal error(18).
 INS1019: System Installation detected an internal error(19).
 INS1020: System Installation detected an internal error(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. Retype a number
 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.

K.3 IFSDEL CID Return Codes

0xFE00 Success
 0x1600 Invalid Command Line
 Command line contains unsupported parameters; either your
 target or the CONFIG.SYS file does not exist.
 0x0800 Data Resource Not Found
 Return code is provided if either the target files could not
 be found or the statements in the CONFIG.SYS could not
 be deleted.
 0x0808 Data Resource Access
 Access to CONFIG.SYS is denied.
 0x1604 Unexpected Condition
 Fatal error during execution attempting an OS/2 call.

K.4 Architected CID Return Codes

Return codes in the range **00 00** to **FC xx** are the install program final return codes with xx varying from 00 to FF.

Return codes in the range **04 00** to **04 FF** are reserved for product specific return codes.

Return codes in the range **FD 00** to **FF xx** are the install program return codes with xx varying from 00 to FF.

The valid return codes and their descriptions are as follows:

Return Code	Description
00 00	Successful program termination
00 04	Successful program termination - Warning Messages Logged - No Reboot
00 08	Successful program termination - Error Messages Logged - No Reboot
00 12	Successful program termination - Severe Error Messages Logged - No Reboot
08 00	Data resource not found
08 04	Data resource access denied because already in use
08 08	Data resource access denied because missing authorization
08 12	Data path not found
08 16	Product not configured
12 00	Storage medium exception (I/O error)
12 04	Device not ready
12 08	Not enough disk space
16 00	Incorrect program invocation
16 04	Unexpected condition
Return Code	Description
FD 00	Reserved return codes
FE 00	Successful program execution - Reboot and don't call me back

FE 04	Successful program execution - Warning Messages Logged - Reboot and don't call me back
FE 08	Successful program execution - Error Messages Logged - Reboot and don't call me back
FE 12	Successful program execution - Severe Error Messages Logged - Reboot and don't call me back
FF xx	Successful program execution - Reboot and call me back

Appendix L. The SERVICE.INI File 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 settable 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.

Name=nnnnnnnn Specifies the NetBIOS name of the SRVIFS code server.

This is the parameter that relates to the *name* specified in a SRVATTCH statement in the SRVIFS redirector's CONFIG.SYS file.

Valid values are 1-8 **alphanumeric** characters if NTS/2 SRVIFS is used. For MPTS 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 = nnn 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 = nnnn 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 = nn 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.

For example, if you have:

1. A fast processor clockspeed (20 MHz or above)
2. 80 client workstations
3. A Token Busmaster adapter

then 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 redirector's.

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 servername 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 an SRVATTCH X: CIDSRV and Path = D:CID and PerClient is set to YES, then for this client D:CIDREQ1 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 name as a subdirectory descendant of *Alias_path*. If subdirectory *Alias_path*"SRVIFS redirector name" does not exist it will be created.

Valid values are Single or PerClient. There is no default value.

Alias_name = nnnnnnnnn

Alias_name is the parameter that relates to the servernameAlias_name specified in a SRVATTCH statement in the SRVIFS redirector's CONFIG.SYS file. There the *servername* 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. For examples see L.2, "The Use of Redirected Drives with Aliases" on page 612.

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 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 name by a period (.). No other characters, including spaces, may be included.

It is also possible to specify an SRVIFS redirector *Name* as asterisk (*) followed by a LAN address to connect regardless of its SRVIFS redirector 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.

Example AuthList file :

Authorization list example

```
*      Start of Sample authorization list
*
#
CLIENT1      SRVIFS redirector name
!
CLIENT2.A1F6955A0010 SRVIFS redirector name
#           with address supplied
*.100005876543 wildcard SRVIFS redirector
#           name with address supplied
#
*      End of Sample 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 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.

```

* SERVICE.INI file used by SERVICE.EXE
;
!
@
#
Adapter = 0 1
MaxClients=25 2
MaxFiles = 102 3
Name=CIDSRV 4
GroupName=No 5
ClientWorkers=12 6
Path=D:\CID 7
PerClient=No 8
PermitWrite = No 9
alias= ReadWrite,Single,log,d:\cid\log 10
alias= ReadOnly,Single,tools,d:\os2tools 10
* Here ends the SERVICE.INI file

```

Notes:

- 1 Number of token-ring adapter being used.
- 2 Adapter 0 is set to support a maximum of 25 concurrent SRVIFS redirector's.
- 3 Maximum of 102 files can be opened concurrently.
- 4 Name of the SRVIFS code server.
- 5 "No", means that the code server name **must** be unique.
- 6 Maximum number of THREADS used to support SRVIFS redirector's requests.
- 7 Fully qualified path to default SRVIFS code server directory.
- 8 "No" PerClient option.
- 9 Read access to default SRVIFS code server directory.
- 10 Alias statement.

Figure 121. Sample SERVICE.INI

L.1 NETBIOS resources

SRVIFS code server and clients are NetBIOS applications and uses NetBIOS resources. If other NetBIOS applications are running on the same machine take care to configure LAPS so that the NetBIOS resources are set high enough. (But do not overdo it, because you do not want to waste memory which could be better used).

Some of the settings in SERVICE.INI affect the required amount of NetBIOS resources as shown below.

Code Server (SERVICE):

Sessions Same value as MaxClients

Commands Number of LAN adapters multiplied by (ClientWorkers+10)

Names 1

For the SRVIFS client there is an optional /S: flag for NetBIOS sessions on the SRVIFS statement in CONFIG.SYS. This is set when THINIFS is executed and /NS is used. **SRVIFS Client:**

Sessions 1 per server it connects to

Defaults to 5

Commands 4

Names 1

L.2 The Use of Redirected Drives with Aliases

Aliases are used to attach a server subdirectory as a drive from a client workstation.

If the alias keyword is used in the following manner:

Alias = ReadOnly,Single,tools,D:\OS2TOOLS

it makes it possible for a client to attach to this server directory (D:OS2TOOLS).

If the server name was CIDSRV, the attach command would read:

SRVATTCH T: \\CIDSRV\TOOLS

thereby accessing the CIDSRV alias named TOOLS. The redirected client T: drive would now have access to all files on D:OS2TOOLS for "read" purposes.

A different type of access would be the PerClient access.

Alias = ReadWrite,PerClient,backup,D:

Assuming the server still being called CIDSRV and the client called CLIENT1, after the following attach command:

SRVATTCH W: \\CIDSRV\BACKUP

The redirected client W: drive would now have read/write access to the D:CLIENT1 subdirectory.

For example a "saveinfo" command file could be created and put in D:OS2TOOLS on the code server. A step could be added to the &SRVIFS. command file which calls the "saveinfo" command file on the T: drive. And the "saveinfo" command file would copy the client's CONFIG.SYS, OS2*.INI files, PROTOCOL.INI and IBMLAN.INI to W:

To make the subdirectory names informative and not have them randomly generated, the IFS statement in the CONFIG.SYS file should either have a real client name, the "*P" option used to prompt for a valid name. This name will become the name of all PerClient subdirectories requested.

The statement:

IFS=A:\SRVIFSC.IFS *P

in the CONFIG.SYS will prompt the user for the client name.

The following figure shows the dependencies of redirected drives.

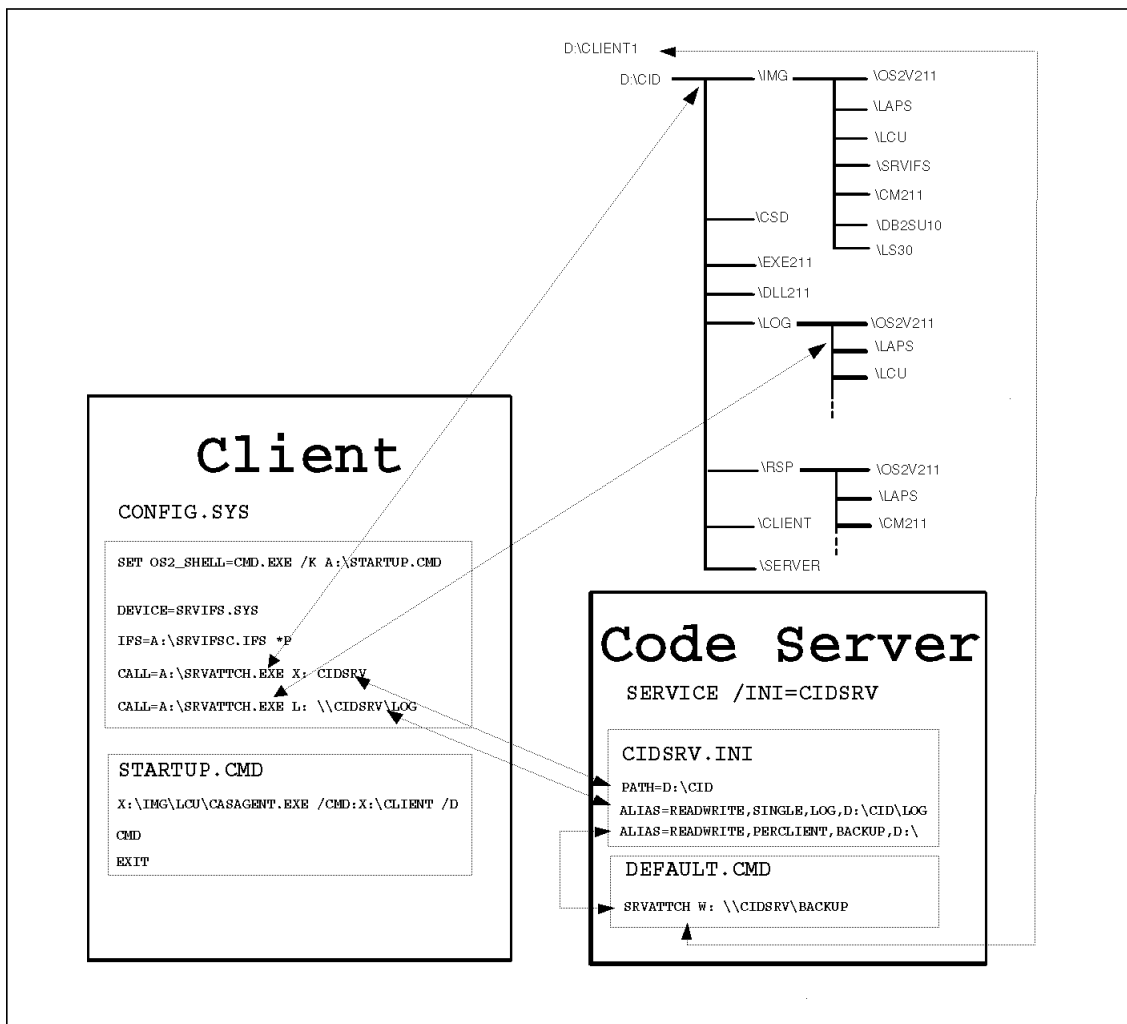


Figure 122. Drive Redirections Using Alias under SRVIFS

This figure shows where the attach commands are processed. The redirected drives X: and L: are attached from the client workstation CONFIG.SYS.

Important

The client workstation will always have redirected drives attached prior to the execution of the LCU command file by placing SRVATTCH statements in CONFIG.SYS of the client workstation.

Drive X: has access in "read only" mode to the code server default alias defined in CIDSRV.INI by:

Path = D:\CID

Note

The default alias is a convenient facility of SRVIFS code server for the alias definitions and is useful if the multiple code servers are used. The recommended attachment technique is a way of exclusive alias definitions in server .INI file.

Drive L: has access in "read/write" mode to the code server LOG alias defined in CIDSRV.INI by:

Alias = ReadWrite,Single,log,D:\CID\LOG

This will result in that files written to L: from CLIENT1 are actually written in D:CIDLOG on the code server.

The redirected drive W: is attached from the execution of the LCU command file on the server. You can always put additional SRVATTCH statements in the LCU command file for redirected drives that do not need to be attached prior to the execution of the LCU command file. See "Additional SRVATTCHs" on page 152 for a more detailed description of additional SRVATTCH statements in the LCU command file.

Drive W: has access in "read/write" mode to the code server BACKUP alias defined in CIDSRV.INI by:

Alias = ReadWrite,PerClient,backup,D:

This will result in that files written to W: from CLIENT1 are actually written in D:CLIENT1 on the code server.

Appendix M. DISKPREP.CMD

A listing of DISKPREP.CMD follows. This program is included on the sample code CDROM in NVDM2EXE. Make sure that the files PIPE.CMD and FDSK*.RSP can be accessed via a redirected drive.

```

/*****
**** File      : DISKPREP.CMD
**** Function   : Automated Partitioning of Harddisks
****
**** Prerequisites:
**** The file PIPE.CMD has to be in the SourcePath. The Input-
**** file for FDISK m u s t create a Partition named OS/2
**** because otherwise the routine will loop.
****
****
****
**** RC 0x0000 = Success:
**** RC 0xFE00 = Success: Reboot Required
**** RC 0x0800 = Error : Data ressource not found
**** RC 0x1200 = Error : IO-Error
**** RC 0x0808 = Error : Access denied, missing authorization
**** RC 0x0812 = Error : datapath not found
**** RC 0x1600 = Error : Incorrect Program invocation
**** RC 0x1604 = Error : Unexpected condition
****
****
*****/

```

```
call CidInit NORXFUNC
```

```

Par.FilePath = ''
Par.ExePath  = ''
Par.SPath    = ''
Ctrl.LogFile = ''
Ctrl.ErrDesc = ''
Par.FDiskMode = ''

```

```
arg Param
```

```

w='FIRST'
i = 1
do until w=''
  w=translate(word(Param, i,))
  if left(w,3)='"/?' then
    do
      call Help
      exit Error.Incorrect_Invocation
    end
  if left(w,3)='"/R:" then do
    w=right(w,length(w)-3)
    Par.FilePath = w
  end /* Do */
  if left(w,3)='"/E:" then do
    w=right(w,length(w)-3)

```

```

        Par.ExePath = w
    end
    if left(w,3)==" /S:" then do
        w=right(w,length(w)-3)
        Par.SPath = w
    end
    if left(w,3)==" /L:" then do
        w=right(w,length(w)-3)
        Ctrl.LogFile = w
    end
    if left(w,3)==" /F:" then do
        w=right(w,length(w)-3)
        Par.FDiskMode = w
    end
    i = i+1
end

ParmsOk = 0
if Ctrl.LogFile=="" then do
    say "Parameter-Error: LogFile missing!"
    ParmsOk = 1
end
else
do
/* You can delete the first part of the log if you want to save space :
        if stream(Ctrl.LogFile, 'c', 'query exists')\=' ' then
            '@del 'Ctrl.LogFile' >Nul' */
        Log='ok'
    end

if Par.FilePath=="" then do
    say "Parameter-Error: Responsefile-Path missing!"
    if Log='ok' then
        call Log "Parameter-Error: Responsefile-Path missing!"
    ParmsOk = 1
    end
if Par.ExePath=="" then do
    say "Parameter-Error: EXE-Path missing "
    if Log='ok' then
        call Log "Parameter-Error: EXE-Path missing "
    ParmsOk = 1
    end
if Par.SPath=="" then do
    say "Parameter-Error: OS2-Source Directory missing !"
    if Log='ok' then
        call Log "Parameter-Error: OS2-Source Directory missing !"
    ParmsOk = 1
    end
if Par.FDiskMode=="" then
    Par.FDiskMode = 'Y'
if ParmsOk == 1 then
    do
        call Help
        exit Error.Incorrect_Invocation
    end

call Log 'DISKPREP'
call Log '-----'

```

```

call Log 'Responsefile Path : 'Par.FilePath
call Log 'EXE-Directory : 'Par.ExePath
call Log 'Source-Directory 'Par.SPath
call Log 'LogFile: 'Ctrl.LogFile
call Log 'FDiskMode: 'Par.FDiskMode
call Log ' '

```

```

Par.PipeIt='| '||Par.ExePath||'\pipe.cmd'

```

```

/* Start of Partitioning Process */

```

```

if NAMECHECK()==1 then do
  if Par.FDiskMode == 'Y' then do
    rc2 = FIRSTFORMAT()
    if rc2 <> 0 then
      do
        call log "Error while formatting volumes.."
        exit Error.IO
      end
    else
      exit Error.Success
    end
  else
    exit Error.Success
  end
else
  do
    call DELETEPART
    rc = Cmd(Par.SPath'\disk_1\fdisk /newmbr') /* New MasterBoot Record */
    call Log 'Query result of Partitioning ...'
    Par.SPath'\disk_1\fdisk /query >>' Ctrl.LogFile
    exit Error.SuccessReboot
  end
  /*****
  NAMECHECK:
  *****/
  /* If a Partition named OS/2 exists this routine returns 1 otherwise 0 */

  call Log 'Checking if Partitioning is already done...'
  Par.SPath'\disk_1\fdisk /query 'Par.PipeIt

  OS2 = 0
  do while queued() > 0
    call GetFDiskLine
    if Name=='OS/2' then do
      OS2 = 1
    end
  end /* Do */
  if OS2==1 then do
    call Log 'OS/2-Partition already exists !'
    return 1
  end
  else
    return 0
  end

```

Help: Procedure Expose Ctrl. Error.

```

say 'DISKPREP'
say "-----"
say "Program-Invocation: "
say "===== "
say ''
say "DISKPREP /R:FilePath /E:ExePath /S:SourcePath"
say "          /L:LogFile /F:Formatmode"
say "With: "
say "      FilePath:      Response-File Directory."
say "      ExePath:       Directory for SETBOOT.EXE and PIPE.CMD."
say "      SourcePath:    Source-Directory (OS/2 Image)."
say "      LogFile:       Logfile."
say "      Formatmode:    Y(es) format Volumes, N(o) do not format."
return

```

GetFDiskLine: Procedure Expose name part drive vtype fstype status ,
start size

```

name = ''
drive = ''
vtype = ''
fstype = ''
status = ''
start = 0
size = 0
if queued() > 0 then do
  pull Line
  drive = strip(substr(Line, 1, 5))
  if datatype(drive, 'N') then do
    name = strip(substr(Line, 7, 10))
    Line = strip(substr(Line, 19))
    part = word(Line, 1)
    vtype = word(Line, 2)
    fstype = word(Line, 3)
    status = word(Line, 4)
    start = word(Line, 5)
    size = word(Line, 6)
  end
end
return

```

/*****
 CheckDrives: procedure expose maxdrives Error. Ctrl. Par. Vol.
 *****/

```

/* Procedure to get the number of installed Harddrives */
call Log 'Checking for number of drives ...'
volumes = XRange('C', 'Z')
maxdrives = 1
maxvolumes = 0
Par.SPath\disk_1\fdisk /query 'Par.PipeIt
do while queued() > 0
  call GetFDiskLine
  if datatype(drive, 'Whole number') then do
    if drive > 0 & drive < 9 then

```

```

        maxdrives = drive
        if pos(left(Part,1),volumes) <> 0 then
        do
            maxvolumes = maxvolumes + 1
            Vol.0 = maxvolumes
            Vol.maxvolumes = Part
        end /* do */
    end /* if */
end /* do */
call Log 'There are 'maxdrives' Drives in the system'
i = 1
do maxvolumes
    call log 'Found volume < ' Vol.i ' >'
    i = i + 1
end /* do */
return

/*****
DELETEPART: procedure expose Error. Par. Ctrl.
*****/

/* Existing partitions will be deleted an a new Partition-tabel is
created with use of response-files:
We handled 2 cases:
a) only one physical Harddrive is installed, so the FDSKHD1.RSP
   is used to create a physical Drive C: , a logical Drive D:
   for Service and ad logical Drive E: for Data

b) there are two physical Harddrives installed, so the FDSKHD2.RSP
   is used to create the same structure as mentioned in a) on the
   first drive two additional logical drives F: and G: on the
   second drive.

If you plan to install a DB2/2-System you should take care of
correct time-settings because otherwise you may run into problems
if the current system-time is set to a future time.
*/

call CheckDrives
count = 1

fstype.0 = 4
fstype.1 = 'FAT'
fstype.2 = 'HPFS'
fstype.3 = 'DOS'
fstype.4 = 'HOA' /* Boot-Manager */

/* Now the existing partitions are selectively deleted to avoid the
deletion of the Reference-partition. */
do maxdrives
    i=1
    do fstype.0 /* for all defined Filesystem-Types */
        cmdline = Par.Spath'\disk_1\fdisk /delete:all /disk:' count,
            ' /fstype:'fstype.i
        i=i+1
        rc2 = Cmd(cmdline)
    end do
end do

```

```

/* Possible Return-Codes of FDISK for successful execution:
1400 = 5120
1C00 = 7168
1E00 = 7680
1800 = 6144
*/

if rc2<>7168 & rc2<>7680 & rc2<>6144 & rc2<>5120 rc2<>0 then do
    call Log 'Error deleting Disk 'count' FSTYPE: 'fstype.i '. Rc: 'rc2
    exit Error.IO
end /* if */
end /* do */
count = count + 1
end /* do */

/* Start of Partitioning: */
/* 1. Boot-Manager */
call Log 'Creating Bootmanager Partition...'
rc2 = Cmd(Par.SPath'\disk_1\fdisk /create /bootmgr /start:t /disk:1')
if rc2 <> 7168 & rc2 <> 6144 then do
    call Log 'Error Creating Bootmanager-Partition. Rc: 'rc2
    exit Error.IO
end /* if */

/* 2. Creating Partitions, dependend on number of Harddrives */

call Log 'Starting Partitioning for 'maxdrives' Harddisk(s) '
rspfile = Par.FilePath || '\FDSKHD\'maxdrives'.RSP'
call Log 'ResponseFile: 'rspfile
rc2 = Cmd(Par.SPath'\disk_1\fdisk /file:'||rspfile)
if rc2 <> 0 then do
    call Log 'Error during Partitioning. Rc: 'rc2
    exit Error.IO
end /* if */

/*****
/* Subroutine to write banner to screen */
/*****
say '●●0;7m';
'@cls'; /* Clear the screen */
Par.SPath'\disk_1\fdisk /query >>'Ctrl.LogFile
say '●●8;1H'; /* Move cursor to line 8 */

do while stream('a:\OS2BOOT', 'c', 'query exists')==''
    call Log 'Waiting for Reentering Boot-Disk !'
    'CLS'
    call PrettyBox 'Disk Partitioning successfully done !',
        'Reenter Boot-Diskette into Drive A: ',,
        'Press ENTER to continue.'
    pull Dummy
end
say "Now the System will be rebootet !!!! "
Par.ExePath'\setboot /b'
do forever
    nop /* waiting forever*/
end /* do */
return /* DELETEPART */
*****/

```



```

FIRSTFORMAT: procedure expose Error. Ctrl. Par. maxdrives Vol.
/*****
rc2 = 0

call CheckDrives
call directory Par.SPath          /* Change to Sourcepath ... */
call directory 'disk_2'          /* ... and to SubDirectory Disk_2 */

/* Now formatting of every drive starts ... */
/* The advantage of doing the FORMAT here ist that the Parameter /L can
   be set to do a fully format. The Format-Options in the OS/2-Response-
   file lead to a quick-format of the partitions. */

volno = 1
do Vol.0 /* Vol.0 = Number of Volumes */
  call log 'Now formatting Drive: 'Vol.volno
  cmdline = 'echo Y | format ' || Vol.volno || ' /FS:HPFS /P /L '
  /* Parameter /P: not documented, suppresses Question for
     Volume-label. */
  rc2 = Cmd(cmdline)
  if rc2 <> 0 then
    do
      call Log "Error formatting "Vol.volno ". Rc: "rc2
      return Error.IO
    end
  else
    volno = volno + 1
  end /* do */

call Log 'Formatting of Harddrives successfully completed.'
return rc2 /* FIRSTFORMAT */

*****/

PrettyBox: procedure
/* Put message in a pretty box */
say ' || left('',76,'') || '||'
do i = 1 to arg()
  say ' || center(arg(i),76) || '||';
end /* do */
say ' || left('',76,'') || '||'
return
/*****/

/* Initialising of Error.- and Ctrl.-Structures */
CidInit: procedure expose Ctrl. Error.
if (Arg(1)\='NORXFUNC') then do
  call RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
  call SysLoadFuncs
end

Error.Success = x2d('0000')
Error.SuccessReboot = x2d('FE00')
Error.Data_Resource_Not_Found = x2d('0800')
Error.Incorrect_Invocation = x2d('1600')

```

```

Error.IO                = x2d('1200')
Error.NoDiskSpace       = x2d('1208')
Error.Access            = x2d('0808')
Error.UnexpectedCondition = x2d('1604')

Ctrl.LogFile = ''
success = 0
return

/* Logging-Procedure: Adds a line to the logfile */

Log: procedure expose Ctrl. Error.

    /* Creating Log-Line-Header */
    success = 0
    header = '•date('E') time() ' DISKPREP'
    text = arg(1)
    if text = '' then
        header = left('', 77, '-')
    rc2 = stream(Ctrl.LogFile, 'c', 'open write')
    if rc2 <> 'READY:' then
        text = 'open' Ctrl.LogFile '=' rc2.' text
    else do
        rc2 = stream(Ctrl.LogFile, 'c', 'seek < 0') /* Set File-Cursor to EOF */
        if rc2 = '' then rc2 = 0
        if \datatype(rc2, 'Whole number') then
            text = 'seek' Ctrl.LogFile '=' rc2.' text
        else do
            rc2 = lineout(Ctrl.LogFile, header text)
            if rc2 <> 0 then
                text = 'lineout' Ctrl.LogFile '=' rc2.' text
            else do
                success = 1
            end /* Do */
        end /* Do */
        call stream Ctrl.LogFile, 'c', 'close'
    end /* Do */
    return

Cmd: Procedure Expose Ctrl.
/* Arg(1) : command */
/* Arg(2) : NOCMDLOG: No errorlog while executing command. */
rc2=0
if Arg(2)=="NOCMDLOG" then
    (Arg(1))
else
    (Arg(1))' 2>>'Ctrl.Logfile /* Stdout and StdErr to Logfile */
rc2=rc
call Log Arg(1). Rc: 'rc2
return rc2

KillQueue: Procedure
do while queued()>0
    pull a
end
return

```

Glossary

ANSI. American National Standards Institute; U.S.-based organization which defines standards for computing devices, protocols, programming languages etc.

Alias name. A redirected drive cannot be accessed directly. An Alias statement on the server points to the server directory or drive, which should be made accessible. Each Alias statement has a name, which must be referred to from a client workstation, when it wants access to this server directory or drive.

API. Application Programming Interface; term used to describe the set of functions by which an application may gain access to operating system services.

Bit. A binary digit, which may be either zero or one. Bits are represented within a computing device by the presence or absence of an electrical or magnetic pulse at a particular point, indicating a one or a zero respectively.

Boot Manager. Feature of OS/2 which allows multiple partitions to exist on fixed disks in the same machine, with a separate operating system on each partition. At boot time, the user may select the desired operating system with which to start the machine.

Byte. A logical data unit composed of eight binary digits (bits).

CD-ROM. Compact Disk Read-Only Memory; technology where data is stored on an optical disk for reading by a computer system equipped with an appropriate reading device. CD-ROM storage media may not be updated by the computer system, although certain implementations allow the media to be erased and re-written.

CID. Configuration, Installation, Distribution. The IBM architected way of automated

installation and customization for OS/2 and other products.

CID code server. LCU server workstation storing images, response files and log information.

CID enabled. CID enabled product can access its product images on the code server. The configuration and installation is done via response file. The product's installation program, which interprets the response file, leaves CID standard return codes.

CID standard return codes. The standard return codes are used to identify the product's installation program behavior under the master installation program execution. These codes identify, for example, the boot and call-back requests.

CSD. see Corrective Service Diskette

CSF. see Corrective Service Facility

Corrective Service Diskette. To maintain OS/2 operating systems, CSDs are supplied, which can be installed using the CSF.

Corrective Service Facility. A mechanism of servicing the OS/2 product line.

DDE. Dynamic Data Exchange; interprocess communication protocol used by applications to define dynamic links. Information updated in one application is automatically reflected in other applications linked to the first application via DDE.

Debugging. The process of removing "bugs" or errors from application code.

Device Driver. Code which handles the translation of generic device commands into specific commands for the required physical device and vice versa, allowing operating

system interaction with physical devices attached to the system.

DLL. Dynamic Link Library; application module containing routines and/or resources, which is dynamically linked with its parent application at load time or runtime rather than during the linkage editing process. The use of DLLs enables decoupling of application routines and resources from the parent program, enhancing code independence, facilitating maintenance and reducing resident memory consumption.

DMA. Direct Memory Addressing; technique by which transfers to and from system memory are made by an independent control chip rather than by the system's main processor, thereby resulting in improved overall performance.

DOS. Disk Operating System; generally used in reference to IBM DOS, the single-tasking 16-bit real-mode operating system designed for Intel 8086 processors, and developed by Microsoft Corporation as MS DOS in the early 1980s. IBM subsequently licensed MS DOS for use on IBM Personal Computer and Personal System/2 machines, and has since undertaken joint development of later versions of the operating system in conjunction with Microsoft.

DOS settings.. Function provided by the Multiple Virtual DOS Machine component of OS/2 which allows a user to customize the behavior of a virtual DOS machine to suit the application running in that VDM. Settings may be configured once by the user and saved, or applications may provide their own configuration information which is used by the VDM upon startup.

DPMI. DOS Protected Mode Interface.

EMS. Expanded Memory Specification; term used to describe the standard developed by Lotus, Intel and Microsoft for access to expanded memory by real mode 80x86 applications.

Expanded Memory. Memory in 80x86 processors, typically on special hardware

adapters, which is accessed by real mode 8086 applications using the LIM EMS specification. Up to 32MB of expanded memory are supported by EMS Version 4.0.

Extended Attributes. Under OS/2 extended attributes are used to provide additional information on files, programs and drivers. Under HPFS the extended attributes are stored together with the file. For file systems, not supporting extended attributes, the EAUTIL program can be used to extract the extended attributes from and storing them in a separate file, as well as reconstructing the original file with extended attributes.

Extended Memory. Memory in 80286, 80386, and 80486-based machines which is located above the 1MB address boundary and accessed using the LIMA XMS specification.

FAT. File Allocation Table; term used to describe the file system implemented by DOS and also supported by OS/2. This file system uses a file allocation table to contain the physical sector addresses of all files on the disk. The FAT file system is supported by OS/2, along with the newer HPFS and other installable file systems.

GB. Gigabyte; 1024 Megabytes, or 1024 x 1024 x 1024 bytes.

HIMEM.SYS. The Extended Memory Manager in general use for DOS.

HPFS. High Performance File System; file system first implemented under OS/2 Version 1.2, offering enhanced performance over the original FAT file system implemented in DOS and prior versions of OS/2. HPFS is an optional installation item under OS/2; the FAT system may also be used to retain compatibility with DOS.

Included response file. The keyword Include of the response file makes it possible to include another response file in a response file, thereby overriding keywords, previously defined before the include statement.

I/O. Input/Output; term used to collectively describe the techniques and devices through which a computer system interfaces with storage devices, external systems and the user.

KB. Kilobyte; 1024 bytes.

LAN. Local Area Network: term used to define a group of devices (typically programmable workstations but also including midrange and host systems) known as nodes, which are located in close geographical proximity to one another (typically within a single property boundary), and which are connected in order to share and exchange information. Typical local area networks include the IBM token-ring network.

LANRES/VM. Local Area Network Resource Extension and Services/VM is an IBM product that provides services to NetWare clients by using Virtual Machine (VM) resources.

LDU. LAN Download Utility: a NetBIOS utility for distribution of software across a LAN supplied by NetView Distribution Manager/2.

LTS. The LAN transport system is used to establish a NetBIOS communication, basic vehicles are needed. The package of programs required to start a successful NetBIOS communication are called LTS.

MB. Megabyte; 1024 Kilobytes, or 1024 x 1024 bytes.

Multiple Virtual DOS Machine. Feature of OS/2 which enables multiple DOS applications to execute concurrently in full screen or windowed mode under OS/2, in conjunction with other 16-bit or 32-bit applications, with full pre-emptive multitasking and memory protection between tasks. See also virtual DOS machine.

NDM/2. NetView Distribution Manager/2: workstation product which interact with NetView DM on the host to provide change management functions.

NetBIOS. NetBIOS stands for Network Basic Input/Output System for LAN. It is an Application Programming Interface (API) that allows high level communication programming on a LAN.

Novell NetWare. Novell NetWare is a network operating system.

Page. Granular unit for memory management using the 80386 and 80486 processors. A page is a 4 KB contiguous unit of memory, which the processor manipulates as a single entity for the purpose of memory manipulation and swapping.

PC Support/400. PC Support/400 is the premier client/server offering for the AS/400 system and the premier cooperative processing application enabler for the AS/400 system.

Physical Device Driver. Protected mode device driver used by the OS/2 operating system and protected mode processes to access hardware devices. DOS applications running in VDMs do not directly access physical device drivers; virtual device drivers are utilized by these applications, and the virtual device driver in turn communicates with the physical device driver.

POST. Power-On Self-Test; code typically stored on ROM (although the IBM PS/2 Model 90 and 95 allow POST code to be stored on fixed disk) which is invoked when a machine is powered on, in order to test the hardware.

Protected Mode. Mode of operation for the Intel 80286 and 80386/80486 processors, whereby the address space is expanded to 16 MB (80286) or 4 GB (80386/80486), and memory references are translated via segment selector and offset, enabling full memory protection between processes executing in the system. With the 80386/80486, paging is available in protected mode.

RAM. Random Access Memory; term used to describe memory which may be dynamically read and written by a processor or other device during system operations. RAM is typically

used to store program instructions and data which not being operated upon by the processor at the current moment in time, but which are required for the logical unit of work currently being carried out.

RAS. Reliability, Availability and Serviceability of the OS/2 operating system.

Real Mode. Default mode of operation for the Intel 80286 and 80386/80486 processors, and the only mode of operation for the 8086 processor. In real mode, the processor acts as a 16-bit device, its physical memory address space is limited to 1 MB, and memory references translate directly to physical addresses. With the 80386 and 80486, paging is not supported in real mode.

Redirected drive. A drive letter, which is not pointing to a local logical or virtual drive, but to a drive or directory on a server.

Response File. The response file is a man-readable ASCII file, prepared in advance, to answer all questions asked during the installation or maintenance of an OS/2 operating system by means of keywords. This file is used during the remote installation or maintenance process.

REXX. Restructured Extended Executor: procedural language originally developed for VM/CMS, which conforms to the SAA standards for procedural languages as defined by SAA *Common Programming Interface Procedures Language Reference*, SC26-4358.

RIPL. Remote Initial Program Load, the booting of a workstation from a server

ROM. Read-Only Memory; term used to describe memory which may be read, but not written to, during system operations. ROM is typically used to store basic hardware initialization instructions, BIOS or self-testing code, which is required to be available prior to accessing the disk subsystem.

SAA. System Application Architecture: set of defined rules, guidelines, interfaces and protocols for application and system design, intended to facilitate cross-system consistency and application portability.

Seed system. The minimum OS/2 operating system booted in order to upgrade the existing operating system on the client workstation.

Service.INI file. This readable ASCII file is needed to start a LCU code server using SRVIFS as a LAN Transport system. Keywords are used to prepare all information required.

Service Pak.. A set of corrective service diskettes supplied to maintain the OS/2 operating system.

TCP/IP. Transmission Control Protocol/Internet Protocol. Provides the flexibility for network interconnection between different systems.

VDM. See Virtual DOS Machine

Virtual DOS Machine. A protected mode process under OS/2 which emulates a DOS operating system environment, such that DOS applications executing within the virtual machine operate exactly as if they were running under DOS. DOS virtual machines support both text and graphics applications. VDMs make use of the virtual 8086 mode of the 80386 and 80486 processors.

Virtual 8086 Mode. Mode of operation of the Intel 80386 and 80486 processors, which allows the processor to execute multiple concurrent tasks with each regarding the processor as its own distinct 8086 processor. This mode of operation provides full pre-emptive multitasking and full memory protection between the virtual 8086 tasks. Also known as V86 mode.

WLFS/VM. Workstation LAN File Services/VM uses VM DASD to provide file sharing services through LAN servers to workstation users.

80386. Intel 80386 microprocessor; the 32-bit processor upon which the OS/2 operating system is based.

80486. Intel 80486 microprocessor; a 32-bit processor which implements a superset of the 80386 processor instruction set.

List of Abbreviations

<i>r/o</i>	read only	<i>IPL</i>	Initial Program Load
<i>r/w</i>	read/write	<i>LAN</i>	Local Area Network
<i>CC</i>	Change Control	<i>LAPS</i>	LAN Adapter and Protocol Support
<i>CDM</i>	Change Distribution Manager	<i>LCU</i>	LAN CID Utility
<i>CID</i>	Configuration, Installation, Distribution	<i>NTS/2</i>	Network Transport Services/2
<i>CM/2</i>	Communications Manager/2	<i>MMPM/2</i>	Multi Media Presentation Manager/2
<i>CSD</i>	Customer Service Diskette	<i>MPTS</i>	Multi-Protocol Transport Services
<i>DB2/2</i>	DATABASE 2 OS/2	<i>OS/2</i>	Operating System/2
<i>DBCS</i>	Double Byte Character Set	<i>PM</i>	Presentation Manager
<i>IBM</i>	International Business Machines Corporation	<i>RIPL</i>	Remote Initial Program Load
<i>ITSO</i>	International Technical Support Organization	<i>SBCS</i>	Single Byte Character Set

Index

Special Characters

- Syntax, BOOTOS2 186
/R 114

Numerics

000e, exit code (DB2 INSTALL) 119

A

abbreviations 633
acronyms 633
Adapter address, universally administered
TR 313
Adapter Keyword on SERVICE.INI file 606
Adding Products to the LCU Command File 154
Additional procedures and files for RIPL 164
AdditionalPrinters 436
AIX 18
Alias 608, 612
 Drive redirection sample 612
 PerClient option usage 612
AlternateAdapter 437
ANX0210 120
ANX0247 120
ANX0264 120
APAR JR08659 119
APM 437
AS/400 18
AS/400. 18
assignment drive letter 52
assignment, drive letter 52
ATA (hard disk) driver, PCMCIA 576
Attended Installation 20, 121
AuthList 609
Authorization list 312, 313
 Keyword description 609
 Sample 610
 working with The 313
AUTORINGSPEED 578

B

BaseFileSystem 49, 437
boot diskette 578
boot diskettes
 build for pristine installation 356
 NVDMBDSK utility 357
Boot Manager
 Add partition to Boot Manager menu 246
 Add partition to utilize Boot Manager 246
 Change partition name for Boot
 Manager 246
 Default partition for Boot Manager 246
 Make startable for Boot Manager 246
 Restrictions when using 243
BOOTDISK 184
BootDrive 157
BOOTOS2 185
BOOTOS2 - Syntax 186
Build command
 to create change file 176
Build Response Files 48

C

Card Services, PCMCIA 576
CASAGENT 105, 309
CASCKREX 106
CASDELET 107
CASINSTL 101, 307
CASPREP Utility 169
CASSETUP
 functions 403
 requirements 404
catalog section
 in change file profile 174
CDROM 438
change file
 from profile created via dialog 179
 in NetView DM/2 173
 installation on client 364

- change file (*continued*)
 - using build command 176
 - using NetView DM/2 dialog 178
- change file profile
 - catalog section 174
 - create via dialog interface 179
 - file specification section 175
 - global statements 174
 - in NetView DM/2 173
 - install section 174
- change management
 - change file 173
 - change file profile 173
 - data objects in NetView DM/2 V2.0 172
 - in NetView DM/2 171
- CheckBoot 145, 159
- CID
 - Client Installation Control Files 79
 - Concepts (and Terminology) 10
 - Enablement of Products 34
 - History, Concepts and Scenarios 3
 - Response file 47
- CID code server
 - Alias 612
 - Authorization list 313
 - Code server installation 299
 - Displaying status 311
 - Forcing stop 312
 - GETBOOT 397
 - Loading LCU 394
 - Loading OS/2 V2.11 CID utilities 373
 - Loading REXX 398
 - Loading SRVIFS 394
 - PerClient 612
 - Preparation 267
 - Refreshing authorization list 312
 - Running code server (SERVICE.EXE) 310
 - Sample CID code server installation
 - program 261
 - Security 312
 - SERVICE.INI customization 315
 - Starting a code server 311
 - Stopping a code server 311
- CID Directory Structure 39
- CID process
 - corequisite groups 365
- CID-Enabled Installation 8
- client
 - response files 464
- Client Access 551, 558
- Client Access for AS/400 18, 551, 558
- Client hardware requirements 265
- Client Installation Control Files 79
- Client preparation for RIPL 286
- Client software requirements 267
- Client workstation
 - CASAGENT 309
 - CASINSTL 307
 - Common LTS diskettes 314, 328
 - Individual LTS diskettes 314
 - LCU client/redirector 305
 - Second THINIFS See SERVICE.INI parameters
 - in 306
 - SRVREXX 309
 - STARTUP.CMD 309
 - THINIFS 305
 - THINLAPS 302
- ClientWorkers 606
- Cloning 6
- CM Server V4.0 116
- CM/2 response file 52
- CM/2 Response File Reference 52
- CMIMAGE 384
- CMLAN Example 112
- CMRECORD 53
- CMSETUP 108
- code server 13
- Code server software 266
- command interface in NetView DM/2 367
- Common LTS diskettes 314, 328
- Communications Manager/2 52, 108
- Communications Server for OS/2 Warp Version
 - 4.0 116
- compatible, downward 40
- CompNameLen keyword 174
- compression 40
- compression, problems 40
- CONFIG.SYS

- configuration
 - keywords 463
- Connecting a Client for Maintenance 183
- Converged Stack (MPTS) 29
- Copy 442, 471
 - syntax 472
- CountryCode 443
- CountryKeyboard 443
- create change file profile
 - via dialog interface 179
- credit card adapter 576
- CRENVVAR.C 547
- CRENVVAR.EXE 545, 547
 - Samples 546
 - Source 547
 - USE 545
- CSF Response file keywords 191

D

- Database 2 for OS/2 117
- DB2 INSTALL, exit code 000e 119
- DB2, INSTALL 56
- DB2/2 Response File 56
- DB2/2 V1.0 Diskette Images 385
 - LANINST 386
- DB2RESP 56, 117
- DDIDest 445
- DDISrc 445
- Default LCU command file 146
- Default Response File 146, 147
- Default response file name 149
- DefaultPrinter 446
- DEVICE, IBM2SS01.SYS 576
- DEVICE, ICRMU01.SYS 577
- DEVICE, PCMCIA.SYS 576
- DiagnosticAids 446
- Directory Structure Considerations
 - for LCU 39, 40
 - for NetView DM/2 43
- Disk image installation
- Disk space for code server 264
- diskette, boot 578
- Disketteless sequence

- DISKPREP.CMD subroutines 254
- DISKPRP.CMD 617
- DisplayAdapter 446
- Documentation 447
- DOSSupport 447
- downward compatible 40
- DPMI 447
- drive letter assignment 52
- drive letter, assignment 52
- DSPINSTL 82
- Dual Boot

E

- EarlyUserExit 447
- ECHO piping on command line 482
- Example, CMLAN 112
- Examples
 - Alias, how to use 612
 - FDISK /query output 250
 - of Authorization list 610
 - SERVICE.INI file 611
 - SRVIFSC statement (*p) 613
- EXIST, IF (CMD command) 481
- ExistingWindowsPath 447
- exit code, 000e (DB2 INSTALL) 119
- ExitOnError 448
- export option in dialog interface 173, 181
- Extended Services V1.0 28
- ExtendedInstall 448

F

- FDISK
 - Command line interface 247
 - Command parameters of 247
 - Description 244
 - Functions under OS/2 246
 - Restrictor parameters of 248
 - Sample command line output 250
- FDISK description 244
- FDSK*.DAT 252
- FFST/2 67
- FileSpecList section
 - in change file profile 175

First Failure Support Technology/2 67
FLASH memory driver, PCMCIA 576
Fonts 448
FormatFAT 49, 449
FormatHPFS 49, 449
FormatPartition 49, 449
FSERVICE.EXE 189

G

GETBOOT 397
GETOSCID 376
GETREXX 398
global name
 in NetView DM/2 V2.0 172
global statements
 in change file profile 174
glossary 627
group response files 464
GroupName 605

H

Handling of Locked Files 138
Hard disk partitioning
 Introduction 243
 Multiple fixed disks HW specifications 243
Hardware and Software Dependencies 571
Hardware requirements for clients 265
HOSTS file 334

I

IBM PC Support/400 551
IBM SNA Services/6000 18
IBM2SS01.SYS 576
IBMLANLK.EXE 139
IBMTOKCS.NIF 578
ICRMU01.SYS 577
ID 449
IFSDEL 99
Include 449, 470, 471, 476
 syntax 471
INCLUDE Keyword
 in DB2/2 Response Files 58
 in LAN Server Response Files 72

INCLUDE Keyword (*continued*)
 in MPTS Response Files 64
IncludeAtEnd 450, 476
Included response files
 Include individual response files
 (diagram) 476
 Include keyword definition 449
 IncludeInLine keyword definition 450
IncludeInLine 450, 477
Individual LTS diskettes 314
INSTALL (DB2) 56, 117
Install Log
Install Procedure
install section
 in change file profile 174
installation
 keywords 463
Installation by Cloning 6
Installation Diskette for RIPL 287
Installation environment
Installation Modes 20
 attended installation 121
 lightly attended installation 121
 unattended installation 121
Installation of Novell NetWare Workstation for
 OS/2 V2.01 128
Installation Scenarios 24
Installation steps for RIPL 271

J

JR08659, APAR 119

K

Keyword, BaseFileSystem 49
Keyword, FormatFAT 49
Keyword, FormatHPFS 49
Keyword, FormatPartition 49
keywords 467
 configuration keywords 463
 installation keywords 463
 list implementation 474
 product-specific 464
 standard 471

keywords (*continued*)

values 468

L

LAN Adapter and Protocol Support 89

LAN CID Utility 26, 28, 94

LAN Requester/Server 121

LAN Server

response file consideration 72

response files 65, 69

LAN Server Response File 65

LAN Server V4.0

LANINSTR 121

LAN Server V5.0 17

initiating remote installation 121

LANINSTR 121

logging 122

LAN Server/400 19

LANINST 386

LANINST program 65

LANINSTR 121

LANRES 554

LAPS 28, 89

LAPSDDEL 94

LAPSDISK 382

LAPSRSP 58

LAPSRSP.EXE 58

LCU

directory structure considerations 39, 40

recovery capability 211

recovery for major failures 215

recovery recommendations 207

sample REXX command file with

recovery 212

LCU command file 143

BootDrive 157

CASPREP Utility 169

Command file structure 148

Default Response File 150

Default response file name 149

Execution of a Diskette Initiated

Installation 156

First section 148

Global variables 149

LCU command file (*continued*)

LAN Server V5.0 RIPL LCU Command File
Sample 164

MPTS SRVIFS LCU Command File
Sample 164

NetWare LCU Command File Sample 168

NUM_INSTALL_PROGS 151

OVERALL_STATE 154

Product count 151

Product data section 149

Product name 149

Response file directory 149

RunInstall 153

Samples and Skeletons 163

Second section 152

SEINST 158

SRVATTCH 152

State variable name 149

Structure index 149

TCP/IP LCU Command File Sample 167

THINIFS 159

Third section 154

LCU command file structure 148

LCU Overview 144

LCU Reboot 145

LCU Reboot and Callback 146

LCU return codes 144

LCU, boot diskette 578

LCU, LAN 28

letter, assignment drive 52

LFS/ESA 552

Lightly Attended Installation 21, 121

Loading Fix Pak Images 197

locked file device driver 139

locked file driver of NetView DM/2 V2.1 142

locked files 138

LOGFILE keyword on CSF response file 192

logging 122

Logging Facilities 35

M

Maintenance 183

Maintenance Partition 184

- Maintenance using Boot Diskettes 183
- Managed System Services/400 18
- MaxClients 606
- MaxFiles 606
- MESSAGE.DAT 52
- Microsoft Windows
 - See Windows
- MicroSoft Word V2.0 Microsoft Workstation
 - Client Installation 423
- MigrateApplications 450
- MigrateConfigFiles 451
- MINSTALL 86
- MKRDPM Utility (RIPL) 287
- Modem driver, PCMCIA 576
- MoreBitmaps 451
- Mouse 451
- MousePort 451
- MPTS 28, 58, 76, 89
 - MPTSRSP3.RSP 58
 - Response File 58
- MPTS CID utilities
 - Description 293
 - GETBOOT 397
 - GETOSCID 376
 - MPTS 89
- MPTS Upgrade 199
- MPTS, brief history of 28
- MPTS, Configuration Files 58
- MPTS, Converged Stack 29
- MPTS, simple rules 76
- MPTS, Unconverged Stack 29
- MSS/400 18
- Multimedia Support 86
- MultimediaSupport 451
- Multiple fixed disks 243
- Multiple fixed disks HW specifications 243

N

- Name 605
- NetView Distribution Manager/2 30
- NetView DM for NetWare 17
- NetView DM for NetWare. 16
- NetView DM/2 17
 - change file 173

- NetView DM/2 (*continued*)
 - change file profile 173
 - change files via dialog 178
 - command interface 367
 - directory structure considerations 43
 - IBMNVDM2.INI file 354
 - installation parameters 354
 - key functions 171
 - ShareDirA (SHAREA) 43
 - ShareDirB (SHAREB) 43
- NetView DM/2 V2.0
 - code server installation 353
 - global names 172
 - objects 172
- NetView DM/2 V2.0 Client Feature 74
- NetView DM/6000 18
- NetWare LCU command file sample 168
- NetWare requirements 318
- Network Transport Services/2 28
- NFSRFL.CMD 337
- Novell NetWare 16
- Novell NetWare for SAA 17
- Novell Procedures
 - NWDELETE.CMD 131
 - NWICON.CMD 129
 - NWINST.CMD 128
 - NWPREP.CMD 135
 - NWSEED.CMD 134
 - STARTNW.CMD 134
- NTS/2 28
- NTS/2 CID utilities
 - CASAGENT 105
 - CASDELET 107
 - CASINSTL 101
 - Description 89
 - GETBOOT 397
 - GETOSCID 376
 - GETREXX 398
 - IFSDEL 99
 - LAPS 89
 - LAPSDDEL 94
 - LAPSDISK 382
 - Quick reference 399
 - SERVICE 310
 - THINIFS 94

NTS/2 CID utilities (*continued*)

- THINLAPS 92
- THINSRV 299
- NUM_INSTALL_PROGS 151
- NVDM/2 Maintenance Partition 185
- NVDMBDSK 578
- NVDMBDSK utility 357
- NWDELETE.CMD 131
- NWICON.CMD 129
- NWINST.CMD 128
- NWPREP.CMD 135
- NWSEED.CMD 134

O

objects

- in NetView DM/2 V2.0 172
- OptionalFileSystem 452
- OptionalSystemUtilities 452
- Order of Product Installation 127
- OS/2 CID utilities
 - Description 79, 373
 - RSPINST 82
 - SEDISK 377
 - SEIMAGE 379
 - SEINST 79
 - SEMAINT 87
- OS/2 Corrective Service Facility
 - See also* OS/2 Corrective Service Facility
 - FSERVICE.EXE 189
 - IBM Communications Manager/2 Version 1.1 Upgrade 201
 - IBM NetView Distribution Manager/2 Version 2.1 Service Paks 205
 - Installation interrupt 194
 - Installation logging 193
 - Installation method 199
 - Introduction 188
 - LAN Server V4.0 Service Pak 203
 - Loading the images on the server 197
 - OS/2 Warp V3 Fix Pak 196
 - OS/2 Warp V3 Fix Pak Response File 197
 - Overview 188
 - Private Fixes 196
 - Response file 191

OS/2 Corrective Service Facility (*continued*)

- Sample Service Pak response file 193
- Select Pak 195
- Service Pak 195
- Servicing of OS/2 Products 194
- OS/2 Corrective Service Facility response file keywords
- OS/2 Response File 48
- OS/2 response file keywords
 - AdditionalPrinters 436
 - AlternateAdapter 437
 - APM 437
 - AuthList 313
 - BaseFileSystem 437
 - CDROM 438
 - Copy 442
 - CountryCode 443
 - CountryKeyboard 443
 - DDIDDP 444
 - DDIDest 445
 - DDISrc 445
 - DefaultPrinter 446
 - DiagnosticAids 446
 - DisplayAdapter 446
 - Documentation 447
 - DOSSupport 447
 - DPMI 447
 - EarlyUserExit 447, 480
 - ExistingWindowsPath 447
 - ExitOnError 448
 - ExtendedInstall 448
 - Fonts 448
 - FormatFAT 449
 - FormatHPFS 449
 - FormatPartition 449
 - ID 449
 - Include 449
 - IncludeAtEnd 450
 - IncludeInLine 450
 - Keyword summary 433
 - MigrateApplications 450
 - MigrateConfigFiles 451
 - MoreBitmaps 451
 - Mouse 451
 - MousePort 451

OS/2 response file keywords (*continued*)

- MultimediaSupport 451
- OptionalFileSystem 452
- OptionalSystemUtilities 452
- OS2IniData 452
- PCMCIA 453
- PCMCIAOptions 454
- PrimaryCodePage 455
- PrinterPort 455
- ProcessEnvironment 455
- ProgressIndication 455
- RebootRequired 456
- REXX 456
- SCSI 456
- SeedConfigSysLine 457
- SerialDeviceSupport 457
- ShareDesktopConfigFiles 458
- SourcePath 458
- TargetDrive 458
- ToolsAndGames 458
- UserExit 459, 480
- Version 459
- WIN-OS/2Desktop 460
- WIN-OS/2Support 460
- WIN-OS/2TargetDrive 461
- WindowedWIN-OS/2 461
- WindowsInstallSourcePath 459
- WindowsSupport 460

OS/2 V2.0 response file

- how to edit 461
- Keyword description 436
- Keyword format 461

OS/2 V2.1 Service Pak 196

OS2IniData 452

OVERALL_STATE 154

Overview of installation steps for RIPL 271

P

PACK 40

Partitioning

- See Hard disk partitioning

Path 607

PC Support/400 551, 558

PC/3270 for OS/2 V4.1 112

PC/3270 for OS/2 V4.1,INSTALL 112

PCMCIA 87, 88, 453, 576

PCMCIA ATA (hard disk) driver 576

PCMCIA Card Services 576

PCMCIA FLASH memory driver 576

PCMCIA Modem driver 576

PCMCIA Socket Services 576

PCMCIA Super Client Drivers 576

PCMCIA.SYS 88, 576

PCMCIAOptions 454

Peer Services 67

PerClient 607, 612

- Subdirectories 612
- Use in Alias Keyword 608

PermitWrite 607

Personal Communications/3270 for OS/2 112

PIPE.CMD 252

PrimaryCodePage 455

Printer Installation Sample 240

Printer Support 217

PrinterPort 455

pristine installation

- build boot diskettes 356
- NVDMBDSK utility 357

Problem (DB2), Remote logging fails 119

problems, compression 40

procedure

ProcessEnvironment 455

Product count 151

product enablement

- response file syntax 466

Product Installation Order 127

Product name 149

Progress Indication 35

ProgressIndication 455

PROTOCOL.INI 58

PROTSHELL statement

R

RebootRequired 456

Recovery Recommendations 207

Redirected drive

- See Alias

- Redirected Drives 35
- Redirected Installation 11, 13
- Redirector workstation
 - Include individual response files (diagram) 476
- Remote Initial Program Load (RIPL) 17
- remote installation
 - LAN Server/Requester 121
- Remote logging fails (DB2 INSTALL) 119
- Remote Multiple Printer Support 217
- Remote_Install_State 145
- REQDELE1.CMD 165
- REQDL300.CMD 165
- REQUPDAT.CMD 165
- resource map utility 577
- Response file 47
- Response File Configurator 239
- Response file directory 149
- Response File Support 11
- Response Files 11, 35, 48
 - client 464
 - comment lines 466
 - description 463
 - group 464
 - hierarchy 474
 - including 464, 470
 - keywords 463, 467
 - LAN requester 65, 72
 - LAN server 65, 69, 72
 - list implementation 474
 - order of processing 473
 - style 470
 - syntax 466
 - whitespace characters 466
- ResponseFile, BaseFileSystem 49
- ResponseFile, FormatFAT 49
- ResponseFile, FormatHPFS 49
- ResponseFile, FormatPartition 49
- Return Codes 36
- REXX
 - REXX keyword in response file 456
- REXX keyword on response file 456
- RINGSPEED 578
- RINSTPRN program 217
- RIPL client/server solution
 - Code server security 291
 - FIT files 273
 - Installation Diskette 287
 - Manual Installation of RIPL Code Server 273
 - MKRDPM Utility 287
 - Overview 270
 - Overview of installation steps 271
 - Preparation for RIPL clients 286
 - RPLENABL Utility 287
 - Running the RIPL code server 290
- RIPL LCU command file sample 164
- RMPI
 - Backup/Restore of Printer and Job Properties 224
 - Definitions for Remote Printer
 - Installation 218
 - Introduction 217
 - Local Printers and Queues 218
 - Network Queues 219
 - Printer and Job Properties 219
 - RINSTPRN Program 220
 - WIN-OS/2 Printers 220
- RMPI Printer response file keywords
 - AdditionalPrinter 227
 - CommPort 227
 - DefaultPrinter 228
 - DefaultQueue 228
 - NetQueue 229
 - Printer 229
 - PrinterDesc 230
 - PrinterName 230
 - PrinterPort 231
 - Properties 231
 - Queue 232
 - QueueDesc 232
 - QueueName 232
 - QueueProc 233
 - WinPrinter 233
- RMPI_CFG Program 239
- RMTREE.CMD 165
- ROM module for RIPL 270
- RPLENABL Utility (RIPL) 287
- RS/6000 18

RSPINST 82, 375
 RunInstall 153
 Running the RIPL code server 290

S

SA: 52
 Sample Code Diskette
 Sample code server installation program 261
 SB: 52
 SCSI keyword on response file 456
 Second Section of LCU Command File 152
 Security 312
 SEDISK 377, 577
 Seed system 87
 SeedConfigSysLine 457
 SEIMAGE 379
 SEINST 79, 151, 158
 Select Pak 195
 SEMAINT 87, 89, 184, 577
 SerialDeviceSupport 457
 Server workstation
 Alias, how to use 612
 Displaying Status 311
 SERVICE.INI customization 315
 SERVICE 310
 SERVICE keyword on CSF response file 191
 Service Pak 195
 Service Pak response file sample 193
 SERVICE.INI 311, 315
 SERVICE.INI file
 Customization of 315
 Description 605
 Sample SERVICE.INI file 611
 SERVICE.INI Keywords
 Adapter 606
 Alias 608
 AuthList 609
 ClientWorkers 606
 GroupName 605
 MaxClients 606
 MaxFiles 606
 Name 605
 Path 607
 PerClient 607, 608

SERVICE.INI Keywords (*continued*)
 PermitWrite 607
 Single 608
 Shared_Directory 52
 ShareDesktopConfigFiles 458
 SNA Services/6000 18
 Socket Services, PCMCIA 576
 Software Distribution Manager 11, 19
 Software for Code Server 266
 Software Installer 56, 117, 120, 411
 CID Enablement using Software
 Installer 411
 Software Installer, Enabling a New product 416
 Software requirements for clients 267
 SourcePath 458
 SRVATTCH 97, 152
 The use of 612
 SRVIFS 94, 305
 SRVIFS Utility 16
 SRVIFSC Command
 using the *"*P"* parameter 613
 using the names feature of 313
 SRVREXX 309
 Stack, Converged (MPTS) 29
 Stack, Unconverged (MPTS) 29
 Standard Installation (non-CID) 4
 Standard Return Codes 36
 STARTNW.CMD 134
 STARTRFI.CMD 325
 STARTRPL.CMD 285
 State variable name 149
 Structure index 149
 Super Client Drivers, PCMCIA 576
 SVGA Adapters (.DSC Files) 84
 SVGA Support 82
 Syntax
 CASDELET 107
 CASPREP 170
 CMSETUP 109
 DSPINSTL 82
 IFSDEL 99
 INSTALL 112
 INSTALL(DB2) 117
 LANINSTR 121
 LAPSDEL 94

Syntax (*continued*)

- LAPSRSP 61
- MPTS 90
- MPTS CASAGENT 105
- MPTS CASINSTL 101
- MPTSCASCKREX 106
- NTS/2 CASAGENT 105
- NTS/2 CASINSTL 101
- RMPI_CFG 239
- RSPINST 82
- SEINST 80
- SEMAINT 87
- THINIFS 95
- THINLAPS 93

System

- recovery 207

T

- TargetDrive 458
- TCP/IP 18
- TCP/IP installation scenario 330
- TCP/IP LCU command file sample 167
- TCP/IP NFS 16
- TCP/IP requirements 330
- TCP/IP Response File 75
- TCP/IP V2.0 Diskette Images 392
- TCP/IP V3.0 123
- TCP/IP V3.0, INSTALL 123
- TCP/IP V3.0, Parameters 123
- THINIFS 94, 159
- THINLAPS 92, 302, 305
- THINR300.CMD 165
- THINSRV 299
- Third Section of LCU Command File 154
- Token-Ring adapter address 313
- ToolsAndGames 458
- Two boot diskette sequence

U

- Unattended Installation 22, 121
- Unconverged Stack (MPTS) 29
- UNINSTALL 81

Universally administered Token Ring adapter

- address 313

UNIX 18

UNPACK 40

- UNPACK for OS/2 374

- UNPACK for OS/2 V2.11 375

- UNPACK for OS/2 Warp V3 375

Upgrading

User Exits

- EarlyUserExit definition 447

- How to use 480

- UserExit 481

- UserExit definition 459

- UserExit for general use 482

- UserExit 459, 471

- syntax 473

- utility, resource map 577

V

- Version 459

W

- WIN-OS/2Desktop 460

- WIN-OS/2Support 460

- WIN-OS/2TargetDrive 461

- WindowedWIN-OS/2 461

Windows

- WindowsInstallSourcePath 459

- WindowsSupport 460



Printed in U.S.A.

S624-4295-01

