

TME® 10 Software Distribution
for OS/2®



Quick Beginnings

Version 3.1.5

TME® 10 Software Distribution
for OS/2®



Quick Beginnings

Version 3.1.5

Note

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

ISO 9001 Certification

ISO 9001 Certification This product was developed using an ISO 9001 certified quality system.

Certification has been awarded by Bureau Veritas Quality International (BVQI) (Certification No. BVQI - 92053).

BVQI is a world leader in quality certification and is currently recognized by more than 20 accreditation bodies.

Third Edition (October 2000)

This is a major revision of, and obsoletes, SH19-4334-02

This edition applies to Version 3.1.5 of TME 10 Software Distribution for OS/2, program number 5639-B06, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product. This program runs under the Version 3.1.5 operating system.

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

IBM and Tivoli welcome your comments. A form for reader's comments is provided at the back of this publication. If the form has been removed, address your comments to the following address:

Distribution Products Information Development
Rome Tivoli Laboratory
IBM SEMEA S.p.A.
Via Sciangai, 53
00144 Rome
Italy

Fax Number (+39) 6 5966 2077
Internet ID: ROMERCF at VNET.IBM.COM

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

© Copyright IBM Corporation 1993, 2000. All rights reserved.

May only be used pursuant to a Tivoli Systems Software License Agreement, an IBM Software License Agreement, or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of IBM Corporation. IBM Corporation grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the IBM Corporation copyright notice. No other rights under copyright are granted without prior written permission of IBM Corporation. The document is not intended for production and is furnished "as is" without warranty of any kind. **All warranties on this document are hereby disclaimed, including the warranties of merchantability and fitness for a particular purpose.**

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

Contents

Notices	xvii
Trademarks and Service Marks	xviii
About This Book	xix
Who Should Use This Book	xix
How This Book Is Organized	xix
TME 10 Software Distribution Publications	xxi
Notation Used in This Book	xxi
What's New in This Release	xxiii
Support for New Platforms	xxiii
New Pristine Scenarios	xxiii
Complete Platform Support Table	xxiii
Deletion of Pending Requests from Host	xxv
Changes to Statuses Reported by 'stattg'	xxvi

Part 1. Introduction to TME 10 Software Distribution 1

Chapter 1. Overview of TME 10 Software Distribution	3
Using TME 10 Software Distribution in a Network	4
TME 10 Software Distribution Users	9
Change Control Administrator	9
Builder	9
User	10
User Profiles	10
TME 10 Software Distribution User Interfaces	10
Graphical User Interface	10
Software distribution Interface	11
Message Log Interface	12
Software Preparation Interface	12
Command Line Interface	13
Starting and Stopping the Command Line Interface	13
Starting TME 10 Software Distribution	13
Starting the Graphical Interfaces	14
Stopping TME 10 Software Distribution	16
Chapter 2. Setting Up a TME 10 Software Distribution Network	17
Types of Target in a TME 10 Software Distribution Network	17
Types of Local Target that Can Be Defined	17
Server	17
Client	17
Mobile Client	18
User Interface Only Client	19
Types of Remote Target that Can Be Defined	19

Server	19
Intermediate Node	20
Single Node	20
Defining Client Targets Automatically	20
Platforms Supported in a TME 10 Software Distribution Network	21
Client Platforms Supported	21
Remote Platforms Supported	21
Communication Protocols that Can Be Used to Link Nodes	22
How Grouping Targets Can Facilitate Change Control Operations	23
The Number of Targets a Network Can Include	24
 Chapter 3. Preparing Software for Change Control	 25
Software Objects	25
Contents of a Software Object	25
Types of Software Object	26
Naming Software Objects	26
Software Object Platform Dependencies	26
Generic Software Objects	27
CID Software Objects	27
The TME 10 Software Distribution Catalog	27
Installing the Same Software Object on Different Workstations	28
Software Preparation Interfaces	30
 Chapter 4. Using Change Control Operations	 31
Change Control Operations	31
Scheduling Change Control Operations in the Network	32
Distributing Data Files	33
Compressing Data Files	33
Translating Data	34
Data Distribution Functions	34
Installing on Pristine Workstations	34
 Chapter 5. Tracking Network Operations	 35
Keeping Track of Workstation Hardware Configurations	35
Keeping Track of Workstation Software Configurations	35
Software Discovery Information	36
Change Control Status Information	36
Tracking the Progress of Change Control Operations	37
 Chapter 6. Using Security Functions	 39
Defining User Authorization Profiles	39
Defining User Access to Data and Targets	41
Defining Data Security	42
 Chapter 7. Finding and Using TME 10 Software Distribution Information	 43
Information about Messages and Error Codes	43
Viewing Online Information	43
Viewing Online Help	44

General Help	44
Message Help	44
Command Help	45
Printing the Documentation	45
PostScript Format	45
PDF Format	46
Calling IBM Service	46

Part 2. Installing and Configuring TME 10 Software Distribution 47

Chapter 8. Planning	49
Hardware Prerequisites	49
Software Prerequisites	49
Communication Protocols	50
Protocols to Connect to Another Server	50
Protocols to Connect to a Client	50
Protocols to connect to a focal point	51
Server/Client Communications	51
Warp 3.0x to Warp 4.0	51
OS/2 4.5 Warp Server for e-business	51
Communications with a Focal Point	52
Prerequisites on the Host System	52
Chapter 9. Installing an OS/2 Distribution Server	53
Installing an OS/2 Distribution Server Interactively	53
Logging on as the Root User	65
Assigning a Password to the User Root	65
Using the APPC Configuration SmartGuides to Connect Over SNA	66
Configuring TCP/IP for Communication with a Remote Server	68
Configuring at the Remote Server Site	69
Updating Your Installation	70
Updating Your Configuration	71
Chapter 10. Editing the Base Configuration File	73
Chapter 11. Defining Users and Targets	83
Defining Users	83
Defining Targets	83
Automatic Client Registration	89
Chapter 12. Configuring for Communication with MVS	91
Configuring Personal Communications/Communications Server	91
Defining the Local Control Point Profile	92
Defining the Logical Link Profile	92
Defining the Local Logical Unit	92
Defining the Partner Logical Unit Profile	93
Defining the Partner LU Location Profile	93
Defining the Mode Profile	93

Defining the TME 10 Software Distribution TP for Inbound Transmissions	93
Defining the TME 10 Software Distribution TP for Outbound Transmissions	94
Starting the Attach Manager	94
Defining Targets for Communication with MVS	94
Configuring at the NetView DM for MVS Site	95
Configuring NCP and VTAM	95
Physical and Logical Unit Definitions	95
Connection through a 3745 Gateway	95
Connection through a Token-Ring Gateway	97
Direct Connection Using an SDLC Link	100
Logon Mode Table Definition	103
Configuring NetView DM for MVS	107
Chapter 13. Configuring Personal Communications/Communications Server	113
Configuration for a SNA/DS Connection	113
Defining the Local Control Point Profile	113
Defining the Logical Link Profile	114
Defining the Local LU Profile	114
Defining the Partner LU Profile	114
Defining the Partner LU Location Profile	114
Defining the Mode Profile	115
Defining the SNA Server TP for Inbound Transmissions	115
Defining the SNA Server TP for Outbound Transmissions	115
Starting the Attach Manager	115
Configuration for an STS Connection using APPC	116
Defining the Local Control Point Profile	116
Defining the Logical Link Profile	117
Defining the Local Logical Unit	117
Defining the Partner Logical Unit	117
Defining the Partner LU Location Profile	117
Defining the Mode Profile	118
Defining the Personal Communications Default Values	118
Defining the SNA Server TP for Inbound and Outbound Transmission	118
Starting The Attach Manager	118
Configuration for an APPC Connection on a Client	118
Defining the Local Control Point Profile	119
Defining the Logical Link Profile	119
Defining the Partner Logical Unit Profile	120
Defining the Partner LU Location Profile	120
Defining the Mode Profile	120
Defining the Personal Communications Default Values	121
Defining the SNA Server TP for Inbound and Outbound Transmission	121
Starting the Attach Manager	121
Chapter 14. Configuring STS Remote Connection Files	123
Editing STS Connection Configuration Files	123
STS Configuration File	123
STS Connection Configuration File	124

Editing the STS Routing Table	126
Planning a Simple Routing Table	127
Planning a Complex Routing Table	127
Determining Destination Addresses	127
Defining Routes	128
Defining STS Targets	131
Chapter 15. Configuring SNA/DS Remote Connection Files	133
Editing Connection Configuration Files	133
SNA/DS Configuration File	133
SNA/DS Connection Configuration Files	135
Editing the SNA/DS Routing Table	137
Planning a Simple Routing Table	138
Planning a Complex Routing Table	138
Determining Destination Addresses	138
Defining Routes	139

Part 3. TME 10 Software Distribution Scenarios 145

Chapter 16. Software Distribution Scenarios	147
The Software Distribution Environment	147
Distribution Database Recovery Tool	151
TME 10 Software Distribution Database Cleanup Tool	151
Scenario 1: Preparing a Non-CID-Enabled Application in Basic Mode	152
Scenario 2: Installing a Software Object to a Target (Push Installation)	161
Verifying the Results of Installation	166
Scenario 3: Preparing a Non-CID-Enabled Application in Advanced Mode	166
Creating a Software Object Profile Using the Remote Directory Function	166
Creating a Dynamic Software Object	172
Scenario 4: Replicating an Installation with DiskCamera	181
Taking a “Before” Picture	181
Installing the Application at the Preparation Site	184
Taking an “After” Picture	185
Cataloging the Software Object	191
Scenario 5: Installing an Application from the Internet using DiskCamera	194
Downloading the Application from the Internet	194
Running IBM AntiVirus	196
Taking a “Before” Picture	198
Installing the Application on the Disk Camera Drive	200
Taking an “After” Picture	202
Scenario 6: Defining and Authorizing a Client to Install Software Objects	205
Scenario 7: Installing an Application at a Client (Pull Installation)	211
Chapter 17. Setting Up for CID Preparation	215
CID Software Distribution Environment	215
Overview of the Steps	217
Running the Code Server Setup Utility at the Preparation Site	217
Completing the Setup on the Code Server Workstation	224

Uploading Software Images to the Code Server	226
Copying Products on the Code Server	231
Adding Products to the Software Library	232

Part 4. Appendixes 235

Appendix A. Installing TME 10 Software Distribution by Response File . . .	237
Using the Installation Command	237
Response File Layout	239
Response File Keywords	239
Installation Definition Keywords	239
General Parameters Configuration Keywords	241
Distribution Configuration Keywords	241
Enterprise Connectivity Configuration Keywords (Server Only)	242

Appendix B. Implementing Inventory Discovery	247
OS/2 and Windows Inventory Discovery Program	247
Writing an Inventory Discovery Program	248
Mapping to the TME 10 Software Distribution Inventory Files	249
Installation Parameters Inventory File	249
Software Inventory File	249
Hardware Inventory File	250
Defining a Filter for Hardware Keywords	251
Running Inventory Discovery	251
Routing Inventory Information to Upstream Servers	252

Appendix C. Replacing the Quiesce Check	253
What the Quiesce Check Does	253
Implementing Quiesce Checks	253
Example of a Check Script Used for Quiesce Checks	253

Appendix D. Writing Change Control Scripts	255
Creating Pre-Install and Post-Install Scripts	255
Creating Pre-Remove and Post-Remove Scripts	256
Creating Pre-Accept and Post-Accept Scripts	256
Creating Pre-Uninstall and Post-Uninstall Scripts	257
Creating Pre-Activate Scripts	257
Understanding Script Exit Codes	257

Appendix E. Writing User Exits	259
Overview	259
Files Provided	259
User Exits	259
Header Files	260
Generating Local File Names (ss_user_loc_name)	260
Notifying Requests (sx_server_request)	260
Notifying Completion (sx_server_report)	261
Modifying Local File Names (cx_daca_filename)	261

Previewing and Modifying Reports (cx_daca_report)	261
Previewing Requests (cx_daca_request)	262
Exporting Directories (sx_export_dir)	262
Removing Exports (sx_unexport_dir)	263
Mounting File Systems (cx_mount_fs)	264
Unmounting File Systems (cx_unmount_fs)	265
Adding Target Parameters (sx_usrexit_trgcfg)	265
Adding User Parameters (sx_usrexit_usrcfg)	267
Reporting Request Completion (sx_server_report)	268
Reporting Fileset Statuses and Products (cx_vercm)	268
Product Information	269
Status Exchange Area	269
TME 10 Software Distribution Statuses	269
Setting Secure Keys (cx_set_secure_key)	270
Checking Secure Keys (cx_check_secure_key)	271
Checking CRC Numbers (cx_check_crcs)	271
Compilation Method	273
Appendix F. Application Definition File Considerations	275
Using an Existing Response File	275
Generating the Response File	275
Creating an Application Definition File	276
Appendix G. Setting Environment Variables	281
Setting Compression or Decompression at the Local Server or at the Destination	
Target	281
Improving the Performance of the Database Access Mechanism on OS/2 4.0	281
Preventing the Loss of Icons When Performing CID Installations	281
Activating an Immediate Reboot	282
Avoiding a Check of Diskette in the Drive	282
Using TME 10 Software Distribution for OS/2 Client with the Windows NT Server	
Multiprocessor via NetBIOS	282
Disabling a Pre-Request Script Process	282
Collecting Product Trace Information	282
Showing Milliseconds in the Date and Time Field	283
Disabling the User ID and Password	283
Setting the Full Path Name of a Trace File	283
Setting a GUI Retry Time	283
Improving the Performance of the Target History Dialog	283
Setting a Server-to-Server Permanent Connection	283
Index	285

Figures

1.	A Single-server TME 10 Software Distribution network	5
2.	Interconnected domains	6
3.	Software distribution from MVS	7
4.	Software distribution from AIX	8
5.	TME 10 Software Distribution for OS/2 folder	14
6.	Logging on to TME 10 Software Distribution	15
7.	Distribution Server documentation - icon view	43
8.	TME 10 Software Distribution welcome window	54
9.	TME 10 Software Distribution install window	54
10.	TME 10 Software Distribution install - directories window	55
11.	Disk space window	56
12.	TME 10 Software Distribution install - progress window	56
13.	Distribution Configuration Notebook general page	57
14.	Distribution Configuration Notebook distribution page - Server	58
15.	TME 10 Software Distribution Enterprise connectivity parameters	60
16.	Remote server connection window	61
17.	Enterprise connectivity page with remote server added	64
18.	Configuration confirmation window	64
19.	SmartGuides - icon view window	67
20.	Example of an STS Routing Table for TCP/IP routes on an AIX remote server	69
21.	Example of an STS Connection Configuration File for a remote server	70
22.	Installation and maintenance window	71
23.	Example of a distribution server Base Configuration File	82
24.	Defining a TME 10 Software Distribution, Version 3.1.5 for OS/2 node at MVS	94
25.	TME 10 Software Distribution and NetView DM for MVS using a direct link and a 3745	96
26.	NCP macros for Token-Ring connectivity	96
27.	VTAM macros for TME 10 Software Distribution connectivity	97
28.	TME 10 Software Distribution and NetView DM for MVS using a Token-Ring gateway	98
29.	VTAM and NCP macros for connectivity through a gateway	99
30.	SDLC connection to NetView DM for MVS	101
31.	VTAM and NCP macros for connectivity using an SDLC link	102
32.	Logon mode table	103
33.	Sample NDMNODE macro for TME 10 Software Distribution Server	108
34.	Sample NDMTCP macro for TME 10 Software Distribution	110
35.	Example of an NDMCP macro	111
36.	Example of an NDMTP macro	111
37.	Example of an STS Configuration file	124
38.	Example of an STS Connection Configuration File for TCP/IP	125
39.	Example of an STS Connection Configuration File for APPC	126
40.	Example of an STS Routing Table for APPC routes	130
41.	Example of an STS Routing Table for TCP/IP routes	130
42.	Example of an STS Routing Table for APPC, TCP/IP, IPX, NetBIOS	131

43.	Example of an SNA/DS Configuration File	135
44.	Example of an SNA/DS Connection Configuration File for APPC	137
45.	Example of an SNA/DS Routing Table for APPC routes	143
46.	Example of an SNA/DS Routing Table for both APPC and TCP/IP protocols	143
47.	Software distribution environment	147
48.	TME 10 Software Distribution for OS/2 Catalog window	148
49.	TME 10 Software Distribution Agent window	148
50.	Catalog informational message	149
51.	TME 10 Software Distribution for OS/2 Catalog window	150
52.	TME 10 Software Distribution Agent window	150
53.	Catalog informational message	151
54.	TME 10 Software Distribution for OS/2 - icon view folder	152
55.	Software preparation folder	153
56.	Software object profiles window	153
57.	New software profile window	154
58.	MY.Sample - basic configuration: general page	155
59.	MY.Sample - basic configuration: files page	156
60.	Open window	156
61.	MY.Sample - basic configuration: files page	157
62.	File options window	158
63.	New software profile window	159
64.	Software object profiles progress indicator	159
65.	Software object profiles informational message	160
66.	Software objects profile window	160
67.	TME 10 Software Distribution for OS/2 - icon view folder	161
68.	TME 10 Software Distribution for OS/2 logon window	162
69.	TME 10 Software Distribution for OS/2 Catalog window	162
70.	Install Change Files window	163
71.	Scheduling information window	164
72.	Schedule time window	164
73.	Scheduling information window	165
74.	Correlators window	165
75.	TME 10 Software Distribution for OS/2 - icon view folder	167
76.	Software preparation folder	167
77.	Software Object profiles window	168
78.	New software profile window	168
79.	My.Sample.Remote - advanced configuration: general page	169
80.	My.Sample.Remote - advanced configuration: remote directory page	170
81.	New software profile window	171
82.	Software object profiles progress indicator	171
83.	Software object profiles informational message	172
84.	Software objects profile window	172
85.	My.Sample.Dynamic - advanced configuration: general page	173
86.	My.Sample.Dynamic - advanced configuration: files page	174
87.	Dynamic sections definition notebook: general page	175
88.	Dynamic sections definition notebook: conditions page	176
89.	Dynamic sections definition notebook: general page	177
90.	Dynamic sections definition notebook: conditions page	178

91.	MY.Sample.Dynamic - advanced configuration: files page	179
92.	New software profile window	179
93.	Software object profiles progress indicator	180
94.	Software object profiles informational message	180
95.	Software object profiles window	180
96.	TME 10 Software Distribution for OS/2 - icon view folder	182
97.	Software preparation folder	182
98.	Software object profiles window	183
99.	Create a software object profile using DiskCamera window	183
100.	Take a picture of a disk drive window	184
101.	Copying files to the DiskCamera drive	185
102.	Editing CONFIG.SYS	185
103.	Software Object Profile generator window	186
104.	Informational message	187
105.	Create a software object profile using DiskCamera window	188
106.	Software object profiles window with XLATOR added	188
107.	XLATOR - advanced configuration notebook	189
108.	XLATOR - advanced configuration notebook: files page	190
109.	XLATOR - advanced configuration notebook: variables page	191
110.	Software object profile window	192
111.	Software object profiles progress indicator	192
112.	Software object profiles informational message	192
113.	TME 10 Software Distribution for OS/2 catalog window	193
114.	Sample File Manager window	194
115.	Downloading the application	195
116.	AntiVirus window	196
117.	Check system for viruses window	197
118.	Select drives/directories window	198
119.	Create a software object profile using DiskCamera window	199
120.	Take a picture of a disk drive window	199
121.	Issuing the install command	200
122.	FM/2 installation window	201
123.	Software object profile generator window	203
124.	Successful profile creation message	204
125.	Create a software object profile using DiskCamera	204
126.	Software object profiles window with FM2 added	205
127.	TME 10 Software Distribution for OS/2 - icon view folder	206
128.	TME 10 Software Distribution for OS/2 logon window	207
129.	TME 10 Software Distribution for OS/2 targets window	207
130.	New local target window	208
131.	TME 10 Software Distribution for OS/2 targets window	209
132.	TME 10 Software Distribution for OS/2 Catalog window	209
133.	Authorize files window	210
134.	Correlators window	210
135.	TME 10 Software Distribution for OS/2 - icon view folder	211
136.	TME 10 Software Distribution for OS/2 logon window	212
137.	TME 10 Software Distribution for OS/2 Catalog window	212
138.	Install Change Files window	213

139.	Correlators window	213
140.	Logical LAN environment for CID distribution	215
141.	TME 10 Software Distribution for OS/2 - icon view folder	218
142.	Software preparation folder	218
143.	CID software preparation window	219
144.	Code server Setup Utility Notebook	219
145.	Type of redirection setting	220
146.	Code server name setting	221
147.	CID directory alias setting	221
148.	LOGINST directory alias setting	222
149.	IBM LAN server setting	223
150.	Recommended directory structure for CID distribution	225
151.	Upload software images – setup window	227
152.	Upload software images setup – in progress window	227
153.	Upload software images window	228
154.	Upload software images overwrite message window	229
155.	Upload software images message window	230
156.	Upload software images message window after upload of the code images	230
157.	Software configurations window	231
158.	New software window	233
159.	Discovered software package	250
160.	Example of a check script used for quiesce checks	254
161.	Generating a Software Object - the process	276

Tables

1.	TME 10 Software Distribution, Version 3.1.5 Platform Support	xxiv
2.	Products from which TME 10 Software Distribution, Version 3.1.5 can be upgraded	xxv
3.	Client Platforms Supported by TME 10 Software Distribution	21
4.	Remote Platforms Supported by TME 10 Software Distribution	21
5.	Connection and Performance Considerations	24
6.	Software Object Status and Associated Functions	37
7.	Administrator Profile (FNDADMN)	40
8.	Builder Profile (FNDBLD)	40
9.	User Profile (FNDUSER)	41
10.	STS Protocols to Connect to Another Server	50
11.	Protocols to Connect to A Client	51
12.	Base Configuration File Parameters	73
13.	Target Definition Parameters	83
14.	VTAM and TME 10 Software Distribution Parameters, 3745 Connection	97
15.	VTAM and TME 10 Software Distribution Parameters, Token-Ring Gateway Connection	100
16.	VTAM and TME 10 Software Distribution Parameters, SDLC Connection	103
17.	Parameters for Defining the LU Session	103
18.	NDMNODE Variables	108
19.	NDMTCP Parameters	110
20.	STS Configuration File Parameters	124
21.	STS Connection Configuration File Parameters	125
22.	Target Identification Terms	128
23.	Defining Routes in a Routing Table	129
24.	SNA/DS Configuration File Parameters	134
25.	SNA/DS Connection Configuration File Parameters	135
26.	Target Identification Terms	139
27.	Defining a Route in the SNA/DS Routing Table	140

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user.

Tivoli Systems or 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:

IBM Director of Licensing,
IBM Corporation,
North Castle Drive,
Armonk,
New York
10504-1785,
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
P.O. Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks and Service Marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both:

IBM	NetFinity
The IBM logo	NetView
Tivoli	OS/2
The Tivoli logo	OS/400
AIX	Portmaster
NetView	PowerPC
AIXwindows	POWERserver
Application System/400	RACF
APPN	RISC System/6000
AS/400	RS/6000
AT	SAA
BookManager	SystemView
DB2	System/370
IBM	VTAM
MVS/ESA	WIN-OS/2
MVS/XA	

Microsoft, Windows, and the Windows NT Logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both and is used by IBM Corporation under license.

UNIX is a registered trademark of the Open Group in the United States and other countries.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

About This Book

This book describes the general concepts, installation, configuration, and use of TME 10 Software Distribution, Version 3.1.5 for OS/2, hereafter also referred to as TME 10 Software Distribution.

Who Should Use This Book

This book is intended for information systems professionals who require a basic understanding of TME 10 Software Distribution in order to evaluate its suitability for software and data distribution and change control in networks based on the OS/2 operating system.

How This Book Is Organized

This book has the following parts and chapters:

Part 1, “Introduction to TME 10 Software Distribution”

This part includes chapters that provide conceptual information about the product. They are:

- Chapter 1, “Overview of TME 10 Software Distribution” describes the benefits you obtain by using change control functions to manage a network of workstations. It introduces different network topologies supported by TME 10 Software Distribution, outlines product users, and tells you how to start and stop the product.
- Chapter 2, “Setting Up a TME 10 Software Distribution Network” provides the information you need to configure the workstations in your network as TME 10 Software Distribution targets.
- Chapter 3, “Preparing Software for Change Control” describes what must be done to prepare software in the format required by TME 10 Software Distribution.
- Chapter 4, “Using Change Control Operations” describes change control and distribution functions, and introduces how to schedule operations in your network.
- Chapter 5, “Tracking Network Operations” describes the facilities provided by TME 10 Software Distribution to keep track of the hardware and software configurations on workstations in your network.
- Chapter 6, “Using Security Functions” describes the security facilities provided by TME 10 Software Distribution.
- Chapter 7, “Finding and Using TME 10 Software Distribution Information” indicates where you can turn to find additional information required to use TME 10 Software Distribution functions.

How This Book Is Organized

Part 2, “Installing and Configuring TME 10 Software Distribution”

This part provides you with the information you need to install and configure the product. It includes the following chapters:

- Chapter 8, “Planning” identifies the hardware and software requirements for installing a TME 10 Software Distribution, Version 3.1.5 for OS/2 server.
- Chapter 9, “Installing an OS/2 Distribution Server” explains how to install a TME 10 Software Distribution server and how to configure for various connectivity options.
- Chapter 10, “Editing the Base Configuration File” explains how to configure TME 10 Software Distribution after installation.
- Chapter 11, “Defining Users and Targets” explains how to define users and targets.
- Chapter 12, “Configuring for Communication with MVS” explains how to configure prerequisite products for connectivity with a NetView DM for MVS focal point.
- Chapter 13, “Configuring Personal Communications/Communications Server” explains how to configure Personal Communications for various connectivity options.
- Chapter 14, “Configuring STS Remote Connection Files” explains how to configure server-to-server connections to other TME 10 Software Distribution systems.
- Chapter 15, “Configuring SNA/DS Remote Connection Files” explains how to configure SNA/DS connections to other TME 10 Software Distribution systems.

Part 3, “TME 10 Software Distribution Scenarios”

This part of the book includes the following chapters:

- Chapter 16, “Software Distribution Scenarios” presents typical user scenarios for preparing software for distribution, cataloging it, and installing it on client workstations.
- Chapter 17, “Setting Up for CID Preparation” explains how to set up the environment for CID software preparation.

Part 4, “Appendixes”

The manual has the following appendixes:

- Appendix A, “Installing TME 10 Software Distribution by Response File” explains how to perform an unattended installation of TME 10 Software Distribution.
- Appendix B, “Implementing Inventory Discovery” provides a description of the inventory discovery procedure.
- Appendix C, “Replacing the Quiesce Check” describes how the quiesce check can be replaced at a target. It is used to see whether users are logged on at a target before operations are begun.

- Appendix D, “Writing Change Control Scripts” describes change management scripts and how to create them.
- Appendix E, “Writing User Exits” on page 259 describes how to edit the C language source code provided with TME 10 Software Distribution to write user exits to create additional functions on a distribution server.
- Appendix F, “Application Definition File Considerations” on page 275 describes how to create an application definition file for use in CID software preparation.
- Appendix G, “Setting Environment Variables” describes how to set the environment variables.

TME 10 Software Distribution Publications

For conceptual information and installation instructions for the TME 10 Software Distribution family of products, consult the appropriate publications:

- *TME 10 Software Distribution for AIX Quick Beginnings*, SH19-4333
- *TME 10 Software Distribution for OS/2 Quick Beginnings*, SH19-4334
- *TME 10 Software Distribution for Windows NT and Windows 2000 Quick Beginnings*, SH19-4335
- *TME 10 Software Distribution for NetWare Quick Beginnings*, SH19-4341
- *TME 10 Software Distribution for NetWare Command Reference*, SH19-4342
- *TME 10 Software Distribution Clients Installation and Configuration*, SH19-4337

Notation Used in This Book

This book uses the following highlighting conventions in text:

Bold	Bold print indicates choices made from a menu or action bar. It is also used to highlight fields and push buttons on panels.
<i>Italics</i>	Italic print is used for introducing new terms in the text or for emphasis.
Monospacing	Monospacing indicates system messages, special characters, statuses, directory names, user input, and examples.
UPPERCASE	Uppercase letters are used for commands, devices, and file names.
<angle brackets>	Angle brackets are used to enclose the names of variables where you must substitute an appropriate value.

Where a command and its associated parameters are too long to be shown on one line, the symbol “►” at the end of a line means that the next line is a continuation of the command string. When you enter the command, enter it all on one line.

What's New in This Release

Release 3.1.5 of TME 10 Software Distribution contains the following new or changed functionalities:

Support for New Platforms

TME 10 Software Distribution, Version 3.1.5 adds support for the following platforms:

- Windows 2000 (Professional and Server)
- Windows NT 4.0 (Service Pack 5 and 6a)
- OS/2, version 4.5 (Warp server for e-business)
- AIX, version 4.3.x

New Pristine Scenarios

TME 10 Software Distribution, Version 3.1.5 Client can be installed on a pristine workstation in the following environments:

- Windows 2000 Professional
- Windows 2000 Server
- Windows NT 4.0 Server/Workstation
- OS/2 4.5 (Warp Server for e-business)
- AIX 4.3.3

This is in addition to the following pristine installation environments, which are maintained from the previous release:

- Windows 3.11
- Windows 95
- Windows NT Version 3.51
- OS/2 3.0.x (Warp)

Complete Platform Support Table

Table 1 on page xxiv shows details of the platforms on which TME 10 Software Distribution is available. The columns in the table contain the following information:

Server Scratch	Indicates whether the Server software can be installed from scratch. Scenarios describing how to carry out the scratch installations can be found in the relevant Quick Beginnings manuals.
Server Upgrade	Indicates which version of the TME 10 Software Distribution Server can be upgraded, by supplying a reference that can be looked up in Table 2 on page xxv. Scenarios describing how to carry out the upgrade can be found in the README file.
Client Scratch	Indicates whether the Client software can be installed from scratch. Scenarios describing how to carry out the scratch installations can be found in the Client Installation and Customization manual.

Support for New Platforms

Client Pristine

Indicates whether the Client software can be installed on a pristine workstation (i.e. a workstation with no operating system installed). Scenarios describing how to carry out the pristine installations can be found in the Pristine and Migration Scenarios manual or the Installation Scenarios for AIX manual.

Client Upgrade

Indicates which version of which Client software can be upgraded, by supplying a reference that can be looked up in Table 2 on page xxv. Scenarios describing how to carry out the upgrade can be found in the relevant README files.

<i>Table 1. TME 10 Software Distribution, Version 3.1.5 Platform Support</i>						
Platform		Server		Client		
OS	Version	Scratch	Upgrade	Scratch	Pristine	Upgrade
Windows	2000 Professional	Y		Y	Y	
	2000 Server	Y		Y	Y	
	NT 4.0 (SP5 & 6a)	Y	1	Y	Y	5
	NT 3.51	Y	1	Y	Y	5
	98			Y		6
	95			Y	Y	6
	3.11			Y	Y	7
OS/2	3.0x	Y	2	Y	Y	8, 11
	4.0	Y	2	Y		8, 11
	4.5 (Warp server for e-business)	Y		Y	Y	
AIX	3.2.5 - 4.2.1	Y	3	Y		9
	4.3.3	Y	3	Y	Y	9
NetWare	4.11 - 4.2x	Y	4	Y		10

Table 2 on page xxv shows the products (and versions) that can be upgraded to TME 10 Software Distribution, Version 3.1.5; the Reference column refers to Table 1.

Table 2. Products from which TME 10 Software Distribution, Version 3.1.5 can be upgraded

Reference (see Table 1)	Version installed	CSD or Fix Pack installed
TME 10 Software Distribution		
1	3.1.3 Server for Windows NT	XR21923
2	3.1.3 Server for OS/2	XR21923
3	3.1.4 Server for AIX	99/10
4	3.1.3 Server for NetWare	XR21924
5	3.1.3 Client for Windows NT	XR21923
6	3.1.3 Client for Windows 9x	XR21923
7	3.1.3 Client for Windows 3.1	XR21923
8	3.1.3 Client for OS/2	XR21923
9	3.1.4 Client for AIX	99/10
10	3.1.3 Client for NetWare	XR21924
NetView DM/2		
11	2.1	

Deletion of Pending Requests from Host

In the circumstances where TME 10 Software Distribution is executing software distribution requests from a focal point running Tivoli NetView Distribution Manager (NetView DM for MVS) Release 7, the MVS focal point can now issue a request to delete any distribution requests that are waiting to be processed or are being processed at the TME 10 Software Distribution server.

- In the case of a distribution request waiting to be processed, the original request will be deleted, and a report sent to the focal point confirming the deletion.
- In the case of a distribution request that is in execution when the deletion request arrives, the original request will be completed, and a report sent to the MVS focal point confirming the successful completion of the original request; no report concerning the unfulfilled deletion request will be sent.

In the case of nodes in a distribution network that are not running TME 10 Software Distribution, Version 3.1.5 (i.e. older versions of TME 10 Software Distribution or NetView DM/2) the deletion requests from the MVS focal point will be ignored.

This functionality runs in the background with no intervention required by the operator of the TME 10 Software Distribution server.

Note: As a consequence of this new functionality global names starting with `$DELETE.$PENDING` are reserved, and may not be used.

Changes to Statuses Reported by 'stattg'

Changes to Statuses Reported by 'stattg'

The `stattg` command gives details of the status of the agent at the local target. A new parameter has been added to the command to reveal additional information.

In the previous releases, and when used without the new parameter, the command reports these statuses:

<i>Available</i>	Agent running and ready to process a request
<i>Not Available</i>	Agent not running or not accessible
<i>Busy</i>	Agent running a request and not available to process any other request.

There are circumstances in which it is possible for the server to have in its database more than one workstation name for the same agent.

For example, if a workstation has been re-defined to the server for some reason, the operator may have supplied a different workstation name than that originally used, but have used the original hostname. In this event, the agent now has the new workstation name, but the server has both workstation names defined; prior to this release the agent reported itself as being *Available* under both workstation names.

With this release, by using the parameter `-c`, in the event that the agent is *Available* and not *Busy*, the command now returns the status *Unknown* if the hostname of the agent is correct but the workstation name in the status request does not match the workstation name of the agent. Thus, by using the `-c` parameter, polling both workstation names will allow you to identify which is the correct one, as one will return the status *Available* and the other *Unknown*. If the parameter is not used, the original functionality is maintained.

However, before using this parameter you should consider the question of the timing of the `stattg` requests. When an agent receives a `stattg` request it sends the status to the server but is then not immediately available to satisfy another request. This means that a second request, received within, say, one minute of the first request, will return the status *Not Available*. If you are polling two suspect workstation names you should wait for this period before sending the second request.

This also means that if you send a `stattg` request using the asterisk wildcard to obtain the status of all or a group of workstations, the results received will depend on whether the *incorrect* workstation name comes before or after the *correct* one in the server's database:

Incorrect workstation name is polled first

The status of the *incorrect* workstation name will be given as *Unknown*, while the *correct* workstation will give *Not Available*

Correct workstation name is polled first

The status of the *correct* workstation name will be given as *Available* while the *incorrect* workstation will give *Not Available*

Thus, after using the asterisk wildcard with the -c parameter, you should individually poll each workstation name given as *Not Available*, waiting for approximately one minute before issuing each command. Workstations that are genuinely unavailable will report the same status as before; workstations that were unavailable while they were recovering from a previous stattg command will now report their true status.

The full details of the stattg command are given in *TME 10 Software Distribution Command Reference*, *TME 10 Software Distribution for NetWare Command Reference* and *TME 10 Software Distribution for AIX Reference*.

Part 1. Introduction to TME 10 Software Distribution

This part provides you with basic information about TME 10 Software Distribution. Its purpose is to help you understand how to use the product effectively to perform data distribution and change control tasks in your network. It includes:

- Chapter 1, "Overview of TME 10 Software Distribution" on page 3
- Chapter 2, "Setting Up a TME 10 Software Distribution Network" on page 17
- Chapter 3, "Preparing Software for Change Control" on page 25
- Chapter 4, "Using Change Control Operations" on page 31
- Chapter 5, "Tracking Network Operations" on page 35
- Chapter 6, "Using Security Functions" on page 39
- Chapter 7, "Finding and Using TME 10 Software Distribution Information" on page 43.

Chapter 1. Overview of TME 10 Software Distribution

Distributed computer systems are essential to the information technology strategies of many companies. The added benefits resulting from automated work processes, however, bring to the forefront additional concerns—the electronic distribution, installation, and maintenance of the software used to perform these processes. How can base software, applications, related configuration data, and user data files on hundreds, even thousands, of workstations be kept up-to-date, consistent, and maintained in a cost-effective and efficient way?

This manual describes how TME 10 Software Distribution, a client/server systems management product, can help solve this complex problem in your enterprise. With TME 10 Software Distribution you can:

- ***Electronically distribute and install software from a central site***

When a system software component or an application that runs on numerous workstations located at different sites, or even in different cities, has to be updated or upgraded, TME 10 Software Distribution can completely automate the procedure. You can prepare and package software at one workstation, send it to the affected target workstations, and install it automatically.

You can facilitate the administration of software installation and maintenance on large numbers of heterogeneous workstations by taking advantage of TME 10 Software Distribution's dynamic functions. By organizing your target workstations into dynamic groups whose members change according to the criteria you establish, you can selectively administer software. You can also organize the content of software packages dynamically, so as to install only certain files on certain workstations.

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

TME 10 Software Distribution automatically stores a history and inventory record of all the hardware and software installed on each workstation in a network. This means that you are always aware of the configuration of all the workstations in your network, and consequently of the activities that need to be performed to ensure consistency where you require it.

- ***Distribute and collect data***

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

- ***Manage a multiplatform environment***

You can take advantage of the benefits of TME 10 Software Distribution in a multiplatform environment. Workstations running OS/2, Windows 3.11, Windows 95, Windows 98, Windows NT, Windows 2000, or Netware can be

Using TME 10 Software Distribution in a Network

controlled from a TME 10 Software Distribution server. TME 10 Software Distribution servers can also interoperate with servers running different operating systems (such as AIX and Windows NT) in the same network. A NetView DM for MVS Release 7 or TME 10 Software Distribution for AIX site can also act as a central site for software and data distribution.

TME 10 Software Distribution can, therefore, become a key element in ensuring the productivity and efficiency of your workstations and users by providing you with the means to:

- ***Save resources, time, and money***

When you install software manually on individual workstations, the process is time-consuming both for those who perform the installation and for those who work at the workstation. When you use TME 10 Software Distribution to automate software management, *one* person (the administrator) from *one* workstation can update thousands of computers, and plan the installation for a convenient time when it does not interfere with anyone's work schedule. What's more, you can automatically create backups of old software, which can immediately be reinstalled should an installation be unsuccessful.

- ***Improve efficiency***

Controlling *all* the software installed on *all* the workstations means ensuring constant compatibility and consistency across your network. No longer will different versions of software be run on different workstations. Unpleasant surprises of incompatibility between workstations will no longer be a common occurrence.

In this book the functions provided by the product are referred to as software distribution and change control.¹ The same term is used across the TME 10 Software Distribution family of products.

Using TME 10 Software Distribution in a Network

You can use TME 10 Software Distribution functions in many different network topologies, which differ greatly in complexity. In the simplest networks, TME 10 Software Distribution is installed on a workstation that is referred to as the network *distribution server* (server). The other workstations in the network become *distribution clients* (clients) that can work in conjunction with a TME 10 Software Distribution server.

Figure 1 on page 5 shows a simple network with a server and its clients. Clients controlled by a server are referred to as *local targets*. The set of local clients, together with the server that controls them, is known as a change control *domain*.

¹ In TME 10 Software Distribution terminology, *change control* is at times referred to as *change management*.

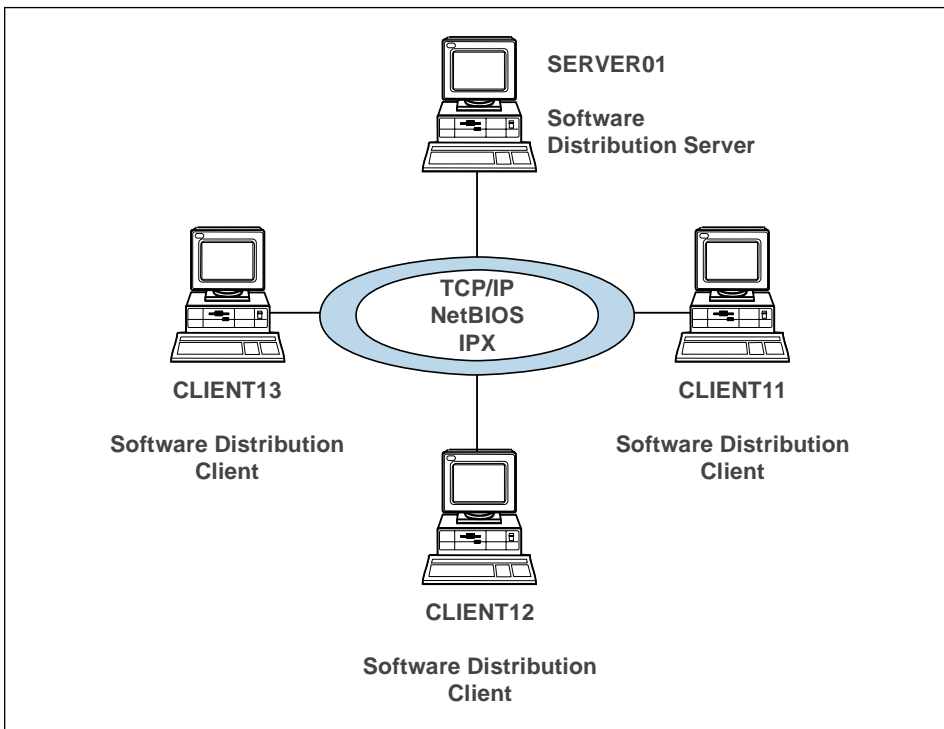


Figure 1. A Single-server TME 10 Software Distribution network

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

More complex networks can combine interconnected domains. Servers and clients in other domains are referred to as *remote targets*. Figure 2 on page 6 shows a network with two domains.

In the figure, SERVER01 can perform data distribution, but not change control, on the clients in DOMAIN02.

Using TME 10 Software Distribution in a Network

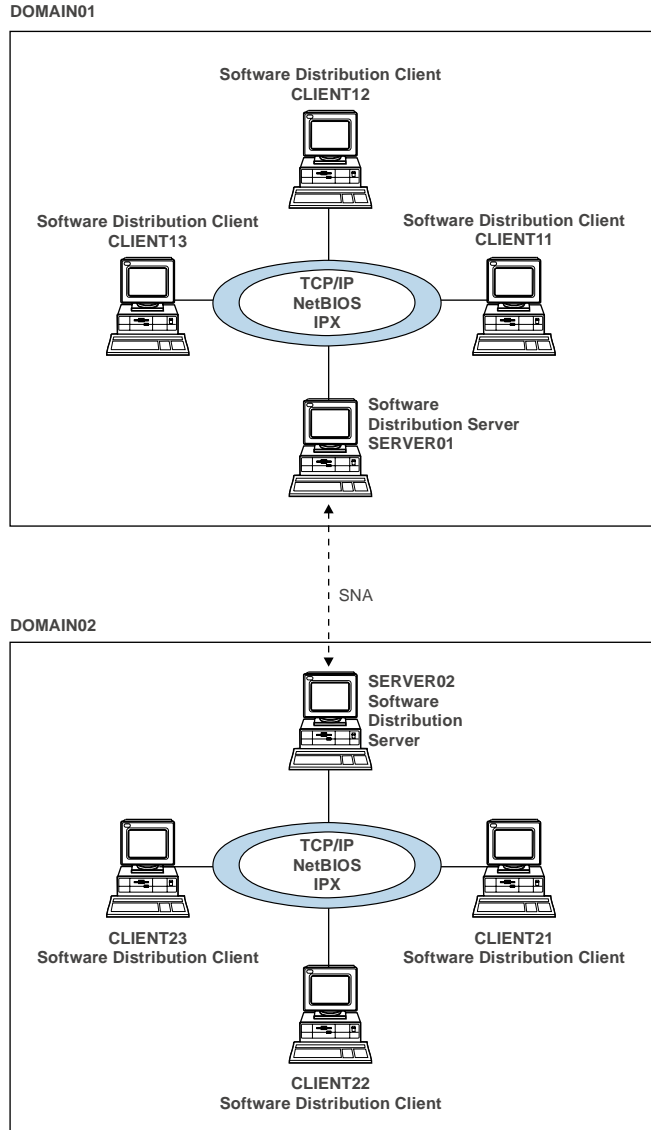


Figure 2. Interconnected domains

Servers can be controlled from one or more central NetView DM for MVS or TME 10 Software Distribution for AIX sites that maintain a total picture of the entire change control network. Figure 3 on page 7 shows a NetView DM for MVS system managing multiple TME 10 Software Distribution domains, and Figure 4 on page 8 shows a similar configuration with a TME 10 Software Distribution for AIX system as the central site.

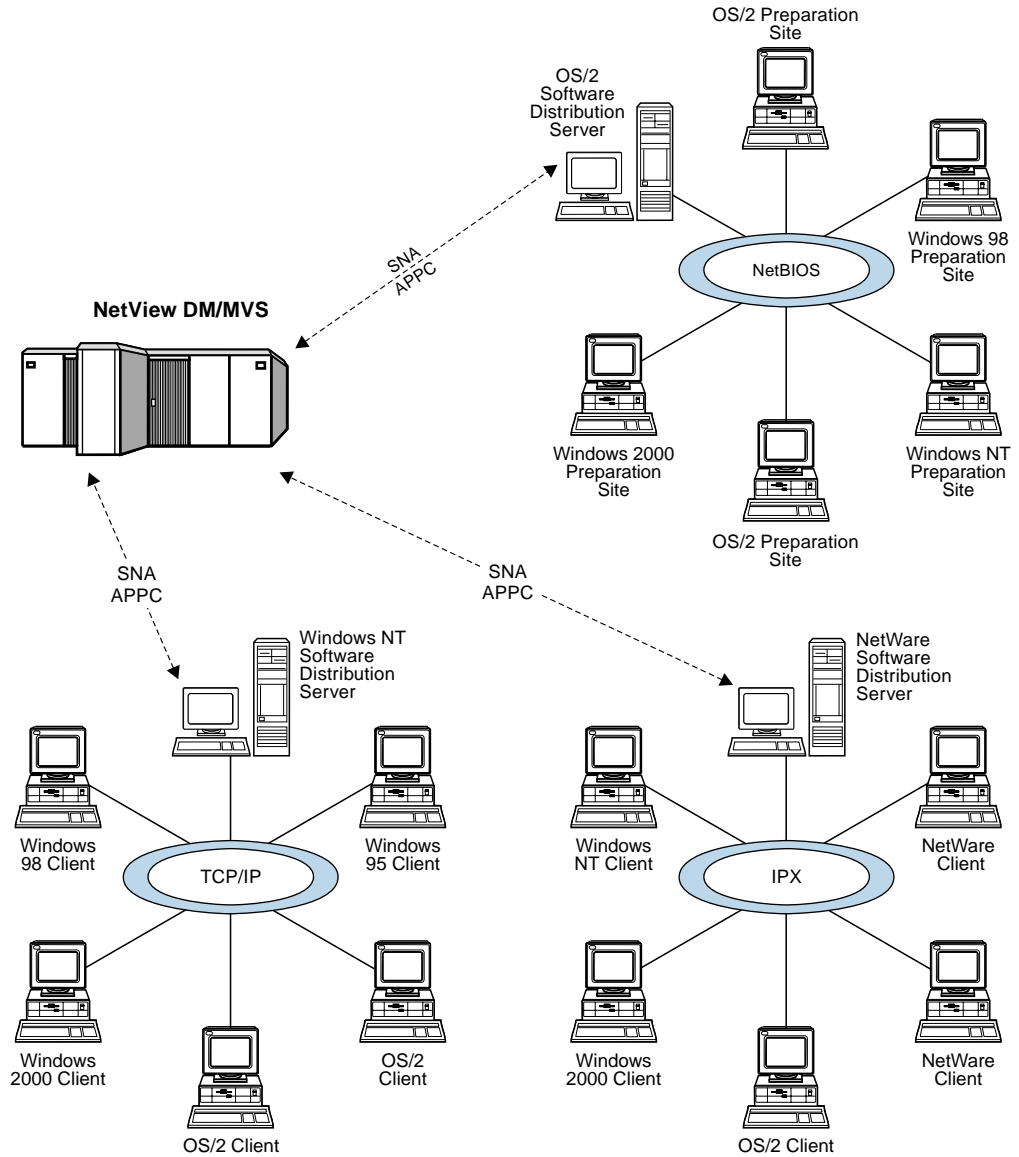


Figure 3. Software distribution from MVS

Using TME 10 Software Distribution in a Network

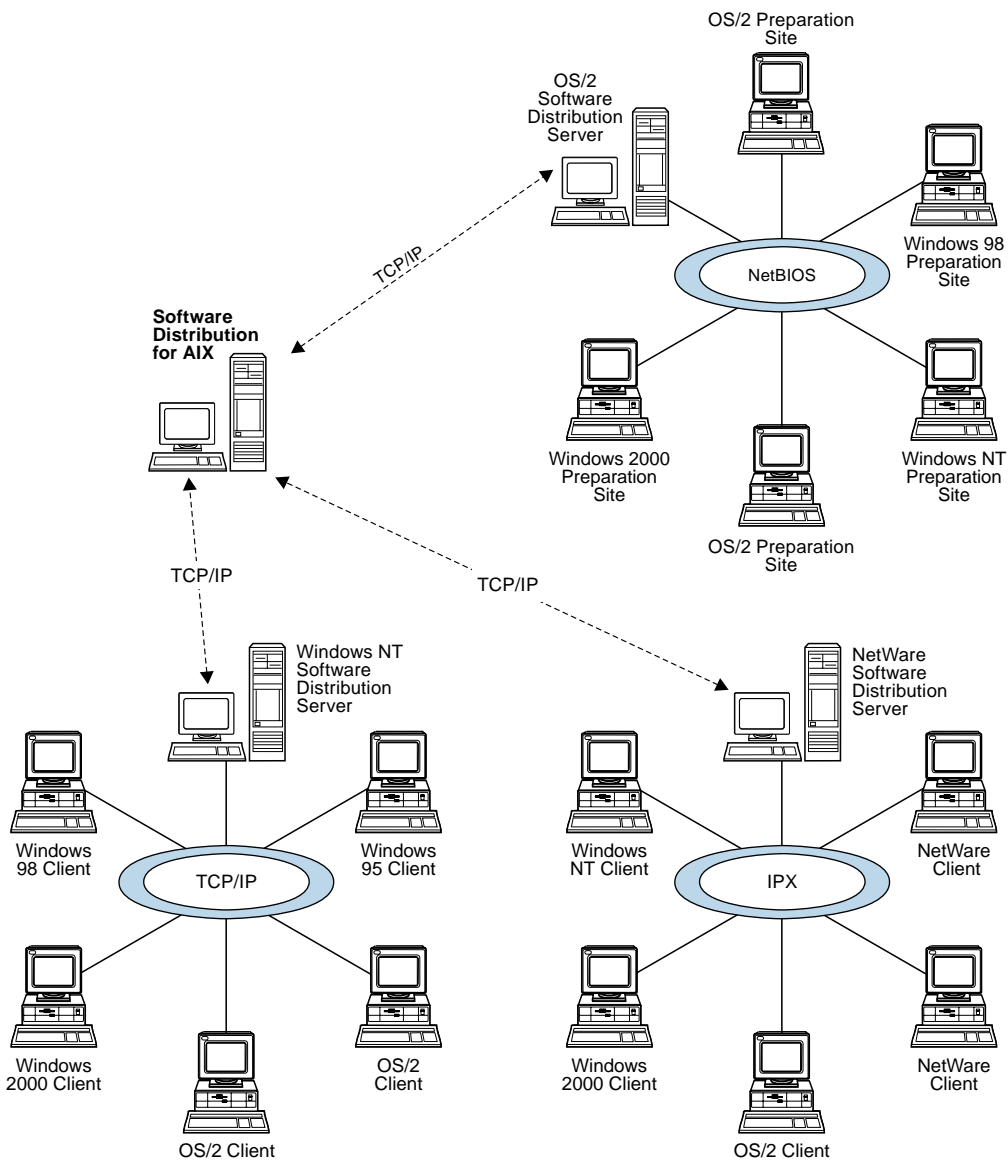


Figure 4. Software distribution from AIX

Networks can also include remote connections to systems running any of the products in the TME 10 Software Distribution/NetView DM for MVS family.

TME 10 Software Distribution Users

Three default classes of users are defined in a TME 10 Software Distribution system:

- Administrator
- Builder
- User

Change Control Administrator

A TME 10 Software Distribution *administrator* is responsible for ensuring that software packages are installed across the network, kept up-to-date, and removed when no longer required. Administrators use TME 10 Software Distribution typically to:

- Define the targets that a server can communicate with, both in its own domain and in remote domains
- Define the users of the product and their authorizations
- Define target, user, and software access authorizations
- Initiate the distribution of data files
- Initiate the managed installation of software packages by the server on client workstations
- Change the software installed on workstations across the network
- View the status of the software packages already installed on client workstations
- Track the status of change control and distribution operations across a network
- Run applications on remote, unattended workstations across the network

In addition to using a server workstation, the administrator can carry out software installation and data distribution functions from any client workstation that is configured in *user-interface-only* (UI only) mode. A UI only client provides an interface to access the server and the capability to initiate change control actions across targets in the network, but no change control operations can be directed to it.

Builder

The *builder* is typically a programmer who is authorized by the change control administrator to prepare the software to be installed using TME 10 Software Distribution. Each separate application or system software product must be prepared as a package suitable for installation by the client program it is destined for.

The builder uses either a server or a client workstation as a *preparation site* for software. The builder can also use a *single node* as a preparation site.

TME 10 Software Distribution User Interfaces

User

The *user* is anyone who uses the services of TME 10 Software Distribution on a client workstation to:

- Perform change control and distribution functions on their own or on other workstations in a network
- Distribute data files across the network to any workstation with a client product installed on it.

User Profiles

User profiles that correspond to these descriptions are installed automatically with the product. However, administrators of the system can create customized profiles that authorize product users to perform any combination of tasks.

TME 10 Software Distribution User Interfaces

You can choose either one of two user interfaces to work with TME 10 Software Distribution:

- Graphical user interface
- Command line interface

Graphical User Interface

The graphical interface provides access to all TME 10 Software Distribution functions. It can be started even if the product is not active, in which case you can perform only a limited number of operations (for example, you can configure new targets but you cannot initiate change control operations).

The graphical interface presents lists of *objects* (such as a list of files) that you select using the mouse or cursor keys. After you select the items you want to work with, select an *action* to perform on them.

The graphical user interface has three components:

- Software distribution
- Message log
- Software preparation

The software preparation interface has two components:

- Generic software
- CID software

Software distribution Interface

This interface is divided logically into main windows from which you access functions. These windows are:

Catalog

This window contains a list of all files currently in the server *catalog*, a database that contains records of all files maintained by the TME 10 Software Distribution system. From this window, you can choose options that enable you to:

- Perform distributions of software objects and data files
- Initiate change control activities
- View change control history records
- Start and stop the product

Targets

Select this window to view a list of the targets that have been set up. From this window you can:

- Create new targets and group targets together
- Modify details about existing targets
- Delete existing targets
- View change control history of targets
- Alter the server configuration

Message Log

This window displays a list of all messages that have been logged. Access it to:

- View messages
- Display help information for any message
- Save the messages to a file

Help

This window can be used as an alternative way to access help information, which can be displayed from any dialog window.

Local Queues

Select this window to display a list of the queues that route requests to clients in a domain. From this window, you can:

- View the contents of queues
- Perform operations on queues

Remote Queues

This window displays a list of the queues that route files to remote domains in a network. From this window, you can:

- View the contents of queues
- Perform operations on queues

This window can be accessed only if the communication option is installed on your system.

TME 10 Software Distribution User Interfaces

Requests

Select this window to track and control the progress of requests that have been submitted to TME 10 Software Distribution. From this window, you can:

- List the status of requests at the request, domain, and target levels.
- Hold, release, reschedule, restart, and delete requests. You can also erase requests from the product database.

From each of these windows, you can access any of the other main windows.

Message Log Interface

You can use this interface to display a list of all the messages that have been logged.

Access it to:

- View messages
- Display help information for any message
- Save the messages to a file

Software Preparation Interface

This interface contains a software preparation component and a CID preparation component.

The first window that appears in the software preparation interface is the Software Preparation window. From this window you can choose whether you want to prepare generic software or CID software.

If you choose generic software, the first window that appears is the Software Object Profiles window. From this window you can access other windows to perform software preparation functions. These windows are:

Catalog

From this window you can perform the following tasks:

- See a list of existing software objects in the catalog
- Group the software objects according to criteria you specify
- Browse the contents of a software object in the catalog

Software profile - create another

From this window you can perform the following tasks:

- Create completely new software objects
- Create new software object profiles from existing profiles
- Create new software objects, using DiskCamera
- Create new software objects from the catalog



The DiskCamera function available in the graphical interface facilitates the creation of software objects by taking pictures of a hard disk before and after a software product is installed on it. The two pictures are compared by DiskCamera, which generates a software object that includes all the new files the comparison

found on the drive as well as any changes that were made to system files as a result of the installation.

If you choose CID software, the first window that appears is the CID Software Preparation window. From this window you can perform the following tasks:

- Set up a code server for images of CID-enabled products and response files
- Upload software images to the code server
- Prepare a CID-enabled product for distribution, generating a response file
- Prepare a CID-enabled product for distribution, using an existing response file

Command Line Interface

The command line interface is composed of commands that you issue from a command prompt. A command is a string of letters with corresponding parameters that you use to make specific requests. Expert users often prefer using the command line as an alternative to the graphical interface because it saves time.

You can perform all TME 10 Software Distribution operations using this interface. For example, to install the software object called `test.file.ref.1.1.2` on a group of targets called `groupone` at 10:00 on March 14, 2000, you would use this line command:

```
nvdn inst test.file.ref.1.1.2 -w groupone -d "14/3/2000" -t "10:00"
```

Line commands and their syntax are documented in the online Command Reference file.

Starting and Stopping the Command Line Interface

You do not have to enter any particular commands to start the command line interface. Simply enter commands from a prompt. Each specific command must be preceded by `nvdn`. For example, to display a list of targets defined in your network, at the `NVDM >` prompt, enter:

```
lstg *
```

Starting TME 10 Software Distribution

You can start TME 10 Software Distribution during the normal startup processing of your system. You can also start and stop the product's components from the desktop, the graphical user interface, or the command line.

To start TME 10 Software Distribution from the desktop, click on the TME 10 Software Distribution icon. The TME 10 Software Distribution folder is displayed, as shown in Figure 5 on page 14.

Starting TME 10 Software Distribution

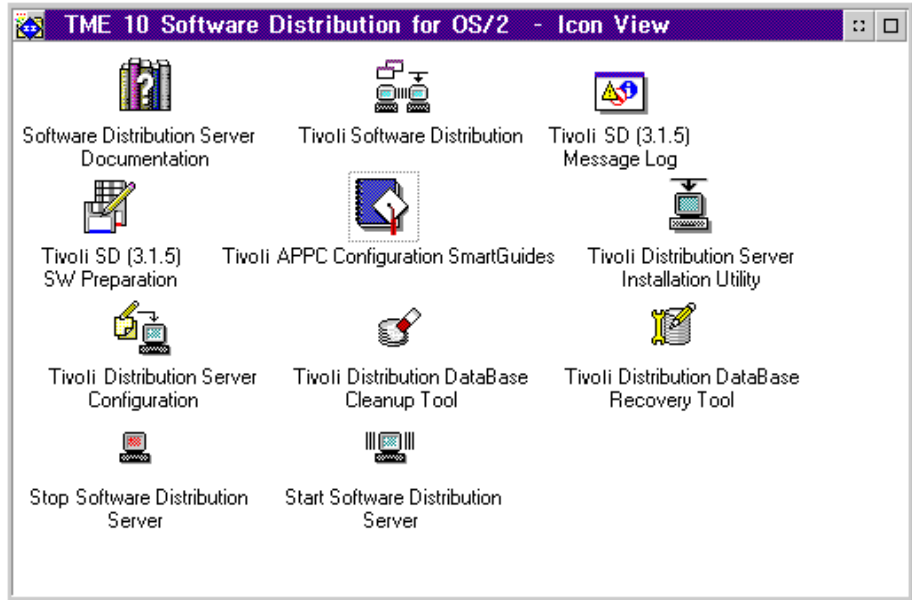


Figure 5. TME 10 Software Distribution for OS/2 folder

Then in the TME 10 Software Distribution folder, click on **Start Distribution Server** or **Start Distribution Client**.

To start TME 10 Software Distribution from the graphical user interface:

1. In the Catalog window in the software distribution interface, select **Engine** from the menu bar and select **Start the system** from the pull-down menu.

A window with the message The system has been started successfully is displayed.

2. Select **OK** to return to the Catalog window.

To start TME 10 Software Distribution from the command line interface, enter the following command from a command prompt:

```
nvdm start
```

Starting the Graphical Interfaces

You can start the graphical interfaces from the desktop or from the command line.

To start the graphical interfaces from the desktop, click on the TME 10 Software Distribution icon. Then in the TME 10 Software Distribution folder, click on the icon for **Software Distribution**, **Software Preparation**, or **Message Log**.

To start the graphical interfaces from the command line, complete the following steps:

1. Enter one of the following commands from the BIN subdirectory in the directory where the product is installed:

- To start the software distribution interface, enter:
nvdmg i
- To start the message log interface, enter: nvdmg i l g
- To start the software and CID software preparation interface, enter: nvdmg i p c

The TME 10 Software Distribution Logo window appears.

2. Select the **OK** push button to continue (If you do not select it, the Logo window will disappear after 10 seconds.). The TME 10 Software Distribution Logon window appears, as shown in Figure 6.

Software Distribution for OS/2 Logon

The password will not be displayed

User ID

Password

Server name

Servers

Figure 6. Logging on to TME 10 Software Distribution

3. Enter your **User ID**, followed by your **Password** if you have one.

Stopping TME 10 Software Distribution

4. In the **Server Name** field enter the name of the server you want to be connected to for this work session. If a list of servers is displayed, select a server from the list.
5. Select the **Logon** push button. The Catalog window is displayed.

Note that starting the graphical interface does not automatically start TME 10 Software Distribution. If the product is not active, follow the instructions given in “Starting TME 10 Software Distribution” on page 13.

Stopping TME 10 Software Distribution

To stop the product from the desktop, click on **Stop Distribution Server** or **Stop Distribution Client** in the product folder.

To stop the product from the graphical interface:

1. In the Catalog window in the software distribution interface, select **Engine** from the menu bar and select **Stop the system** from the pull-down menu.
2. Select **OK** to return to the Catalog window.

The graphical interface stops when all the main windows have been closed. To close a window, do one of the following:

- Press Alt+F4
- Select **Close** from the System menu
- Double-click on the System Menu icon
- Select **Close all** from the Windows menu to close all the windows and stop the graphical interface.

To stop TME 10 Software Distribution from the command line interface, enter the following command from a command prompt:

```
nvdm stop
```

Chapter 2. Setting Up a TME 10 Software Distribution Network

This chapter provides information to help you set up workstations in your network as TME 10 Software Distribution targets, and to configure their communication links. It also describes the functions you can use to keep track of the hardware configuration of the workstations in your network.

Types of Target in a TME 10 Software Distribution Network

Targets are defined in a TME 10 Software Distribution network according to the role they are to perform and how they are to perform that role (specified as target *mode*). This section describes the various roles you can assign to targets.

Targets are either local or remote, depending on their location in the network. A target that is in the same domain as the target communicating with it is a *local* target. When targets perform data distribution across domains in multidomain networks, they are *remote* targets.

Types of Local Target that Can Be Defined

You can define local targets to perform one of the roles described in the following sections.

Server

Servers contain the catalog where software objects and data files, as well as change control history records, are stored. They also contain the configuration information for all the targets they can communicate with, be they local or remote. Whenever a client communicates with another target, the communication is routed through its server.

A server can initiate change control and distribution operations on targets in their own domain. They can also perform distribution operations on remote targets.

Role	Target Type	Target Mode
Initiate change control on targets	Server	Push
Initiate distribution on targets	Server	Push

Client

Client targets work in conjunction with a server, and must have a TME 10 Software Distribution Client product installed. A local client can:

- Be the object of change control operations initiated by a server, when configured in *push* mode
- Distribute data files (send, retrieve, and delete them) to other clients in its domain
- Initiate change control operations on itself, using any software available to it from the server catalog, when configured in *pull* mode

Types of Local Target that Can Be Defined

- Build software objects and store them in the server catalog
- View information about the software installed on its workstation, as well as information about the other clients and software available in the network
- Catalog and uncatalog data files

The following modes can be defined for a client target:

Role	Target Type	Target Mode
Server/other client initiates change control	Client	Push
Initiates change control on self	Client	Pull
Prepare software objects	Client	Push or pull
Send, retrieve, delete data files	Client	Push or pull

Mobile Client

Mobile clients are local targets on which it is possible to do change management operations without connection to a server. Mobile clients must be installed with the Mobile Client component.

You connect a mobile client using the GUI or the **connect** command on the command line. With this command you can also establish the time when the connection window is opened, the connection duration, and a recursive daily connection. You disconnect a mobile client using the GUI or the **disconnect** command on the command line. With this command you can also stop the established recursive daily connection from the client to the server. See the helps or the *Command Reference* for more details.

Because mobile clients are not always connected with a server whose catalog they can access, they have their own local catalog. Once a mobile client's catalog has been updated, the client can perform the following operations locally:

- Install software and software changes
- Remove software
- Accept changes to software
- Activate pending changes on your workstation
- Uninstall software packages and software changes, along with any updates or fixes applied
- Catalog and uncatalog files in the local catalog
- Import software objects from external devices
- View software object information
- View and purge pending requests

Mobile clients are not forced to work locally; they can connect to a server and function as non-mobile clients, performing all the tasks described in “Client” on page 17.

Types of Remote Target that Can Be Defined

Mobile clients can also function as *fully disconnected* clients, meaning that they never connect to a server. They update their local catalog with software objects manually from external media (CD-ROMs, tapes, diskettes), and then perform change control on themselves. Change control status information stored in the server catalog for fully disconnected mobile clients can then be updated manually, using the **updcn** command. A fully disconnected client must set the FULLY DISCONNECTED keyword in its base configuration file to YES.

Define the following attributes for mobile clients:

Role	Target Type	Target Mode
Work as mobile client	Mobile	Push or pull
Work as fully disconnected mobile client	Client	Disconnected

User Interface Only Client

These targets can be used *only* to run the TME 10 Software Distribution user interfaces. This type of target is useful when you have an environment with more than one server. It allows an administrator to access all servers from the same target, either to perform administrative tasks or to schedule operations.

A workstation configured in this manner is used to initiate change control on *other* targets, or to request distributions to and from the server. User interface only targets cannot be the object of change control instructions from the same server they are defined as a user interface only target for.

Define the following attributes for this type of target:

Role	Target Type	Target Mode
User interfaces at server	UI only	None

Types of Remote Target that Can Be Defined

Remote targets can perform a greater variety of roles, in networks with more complex topologies. The possibilities are described in the following sections.

Server

Servers contain the catalog where software objects and data files, as well as change control history records, are stored. They also contain the configuration information for all the targets they can communicate with, be they local or remote. Whenever a client communicates with another target, the communication is routed through its server.

A server can initiate change control and distribution operations on targets in their own domain. They can also perform distribution operations on remote targets.

Types of Remote Target that Can Be Defined

Role	Target Type	Target Mode
Initiate change control on targets	Server	Push
Initiate distribution on targets	Server	Push

Intermediate Node

Remote servers can act as intermediate nodes, whose task is to fan out requests to downstream nodes. The fan-out operation creates multiple copies of the same object to be distributed to a number of, or a group of, targets.

Using intermediate nodes is often an efficient and cheaper way of distributing large quantities of data to numerous destination targets, because they eliminate the need for a direct connection between the target that originates an operation and the targets addressed by the operation.

Role	Target Type	Target Mode
Fan-out change control requests	Server	Push

Single Node

You can use single-node targets as preparation sites for software.

Role	Target Type	Target Mode
Act as preparation site	Single	Push

Defining Client Targets Automatically

You do not have to configure all client targets in a network individually from a server. You can set up your network so that they are configured automatically, or autoregistered, the first time a client target connects to a server. For autoregistration to take place, the AUTOMATIC TARGET REGISTRATION keyword in a server's base configuration file must be set to YES, and the TARGET ADDRESS and TARGET MODE keywords must be specified in the client base configuration file. See Chapter 10, "Editing the Base Configuration File" on page 73 for a description of the server base configuration file, and the *Installation and Configuration* manual for the client base configuration files.

When a client is automatically configured, its address and mode are inserted in its server database. Any other parameters for the client target must be specified manually using the graphical interface or the command line interface.

Platforms Supported in a TME 10 Software Distribution Network

TME 10 Software Distribution networks can include workstations that are connected across local and remote links. As a rule, local links connect servers to their clients and remote links connect servers to other servers in different domains.

Client Platforms Supported

You can connect a server to clients that run any of the following operating systems, provided they have the corresponding TME 10 Software Distribution Client product installed:

Table 3. Client Platforms Supported by TME 10 Software Distribution

Platform	Product
AIX/6000®	TME 10 Software Distribution for AIX Client
OS/2	TME 10 Software Distribution for OS/2 Client
Windows 2000	TME 10 Software Distribution for Windows 2000 Client
Windows NT	TME 10 Software Distribution for Windows NT Client
Windows 95 & 98	TME 10 Software Distribution for Windows 9x Client
Windows 3.11	TME 10 Software Distribution for Windows 3.11 Client
NetWare	TME 10 Software Distribution Client for NetWare

Remote Platforms Supported

You can connect a server to workstations in other domains. As described in “Types of Remote Target that Can Be Defined” on page 19, these workstations can be configured as:

- Servers configured as managers or focal points
- Intermediate nodes
- Single nodes

Remote workstations can run any of the following operating systems, provided they have the corresponding TME 10 Software Distribution product installed. Table 4 lists supported platforms and counterpart products, and the roles they perform within the network.

Table 4 (Page 1 of 2). Remote Platforms Supported by TME 10 Software Distribution

Platform	Product	Role
AIX/6000	TME 10 Software Distribution for AIX	Manager Focal point Intermediate node Remote server (data distribution only)
OS/2	TME 10 Software Distribution for OS/2	Remote server (data distribution only) Intermediate node Single node

Communication Protocols that Can Be Used to Link Nodes

Table 4 (Page 2 of 2). Remote Platforms Supported by TME 10 Software Distribution

Platform	Product	Role
Windows 2000	TME 10 Software Distribution for Windows 2000	Remote server (data distribution only) Intermediate node Single node
Windows NT	TME 10 Software Distribution for Windows NT	Remote server (data distribution only) Intermediate node Single node
NetWare	TME 10 Software Distribution for NetWare	Remote server (data distribution only) Intermediate node Single node
MVS	NetView DM for MVS Release 7	Manager Focal point

Communication Protocols that Can Be Used to Link Nodes

Clients, servers, and TME 10 Software Distribution/NetView DM for MVS family products can be linked in a network using different transmission protocols:

NetBIOS, TCP/IP, IPX, or APPC

To connect a server and its clients.

SNA/DS across APPC

You can configure SNA/DS across APPC for remote server/server connections to another OS/2 server, an AIX server, a NetWare server, or a Windows NT server acting as an intermediate node. You can also connect to a NetView DM for MVS focal point over APPC.

STS across APPC, TCP /IP, NetBIOS, or IPX

STS is the acronym for the term *server-to-server*. This is an internal TME 10 Software Distribution transport mechanism that can be configured in networks connected:

- Over APPC, TCP/IP, NetBIOS, or IPX for remote server/server connections to an OS/2 server or an OS/2 client
- Over TCP/IP, NetBIOS, or IPX for remote server/server connections to a Windows NT server
- Over APPC or TCP/IP for remote server/server connections to an AIX server
- Over IPX or TCP/IP for remote server/server connections to a NetWare server

STS communication offers better performance for many transmission operations, as well as these additional functions:

- You can send, retrieve, and delete data files that are not stored in the catalog, called *uncataloged* files. Uncataloged data files can also be used as procedures to be executed at targets.
- You can authorize data files so that they can be executed at remote targets.

"Communication Protocols" on page 50 provides full details of the communications protocols that can be used from a server with TME 10 Software Distribution, Version 3.1.5 for OS/2 installed to any client or server with which TME 10 Software Distribution can communicate.

How Grouping Targets Can Facilitate Change Control Operations

Grouping targets into homogeneous units can be a useful aid in organizing and scheduling change control operations, especially when you are dealing with an extensive network. When you request change control and distribution operations for a group, the operation is performed on all the targets included in it.

Groups can include up to 1000 local and remote targets, whose mode can be either push, pull, manager, or focal. What's more, you can define *static groups*, whose members are always the same, or *dynamic groups*, whose members change according to criteria you define for populating the group.

A dynamic group can contain both static and dynamic members. The static members are always included in the group, while the dynamic members change according to the filters and rules you establish for populating the group. Filters allow you to define the members of a group according to:

- Target names
- Target types (client, mobile, server, single, UI only)
- Target modes (push, pull, focal point, manager)
- Target operating systems
- Target access keys

Rules are statements that specify the logical relationship between values you specify for any of these tokens:

Target installation parameters

Tokens substituted during the installation process. Installation parameters usually correspond to directory and path names.

Hardware tokens

Hardware requirements specified when targets are defined individually to the system.

Change management statuses

The statuses that form part of the history records stored in the catalog for software objects and targets.

The following examples give you an idea of how you can use static and dynamic groups to your advantage:

- If you must send DATA_A1 to all the targets in EuroTravel's branch offices every week, you can create a static group called BRANCHES, that holds all the targets the file must be sent to. Then every week you simply have to send DATA_A1 to BRANCHES.

The Number of Targets a Network Can Include

- Suppose you have to install a new version of a word processor only on those targets in EuroTravel offices where the word processor is already installed. You can create a dynamic group called HAVE_WORDPROC; the rule used to populate the group would state “include those targets whose change management status is equal to install for the EURO.WORDPROC.REF.0 software object”. Then install the word processor on HAVE_WORDPROC.

The Number of Targets a Network Can Include

When you are planning the topology of a network, you must know how many workstations TME 10 Software Distribution can support, as shown in Table 5.

<i>Table 5. Connection and Performance Considerations</i>	
Maximum number of clients per server	1000
Maximum number of concurrently active clients in a domain	50

Chapter 3. Preparing Software for Change Control

This chapter describes what you must do to the software package so that it can be used in change control operations.

Software Objects

Changes to software on a workstation are distributed in packages that contain software files together with instructions about how they are to be installed. These packages are called *software objects*. (Software objects are also called *change files*.)

Contents of a Software Object

The main information included in software objects is:

- Specifications of the files to be included in the package and their location.
- Specifications of the location of files stored in directories at a *remote site*. These remote directories are mounted from a target when an installation or other change control operation is performed. When you use this remote source method, you can make use of file servers in your network and create smaller software objects that occupy minimum disk space at both the server and clients.
- Specifications of directories to be created or deleted at the target workstation.
- Response files (for packages with CID conformance only; see “CID Software Objects” on page 27).
- Information about hardware and software prerequisites and corequisites, including disk storage space requirements. TME 10 Software Distribution checks for the presence of prerequisite and corequisite software and the amount of disk space available for a package before attempting to install it.

Software objects that include all the prerequisite files required for an installation can be generated automatically.

- Scripts or programs that tailor the installation and other operations to be executed on the target.
- A description of each product or file that is included in the software object. This information is convenient for tracing software object history.
- Compression options and packing instructions to apply if transmitting or storing the software object in compressed format. You can also specify if each file in the software object is to be compressed at build time and decompressed at install time.

Software Object Platform Dependencies

Types of Software Object

You create different types of software object depending on whether you have to install new software, update existing software, or apply fixes to existing software at targets. These types of software object are referred to by the following *change names*:

Refresh (REF)	Software objects that contain a complete new copy of the software or data item being changed to a new level, release, or version.
Update (UPD)	Software objects that contain an update to a component and are installed on top of the original. Updates change the level of the component ID. To install an update, you must first install a refresh software object as not removable.
Fix (FIX)	Software objects that contain a fix for an existing component and are installed on top of the original or an update. Fixes do not change the level of the component ID. To install a fix, you must first install a refresh software object as not removable.

Naming Software Objects

The names you assign to software objects follow the convention for *labels* (also called *global names*). Using labels ensures that each software object is identified uniquely. A label consists of three main elements:

- The name of the component or software package
- The type of change to the component or software package (refresh, update, or fix)
- The level of change to the component or software package.

The following conventions apply when you create a label:

- The name can have from 2 to 10 parts
- Parts must be separated by a period
- The label cannot start or end with a period
- Each part cannot exceed 16 characters
- The label, including periods, cannot exceed 64 characters
- Numbers, uppercase letters, and the characters \$, #, @, and _ can be used in labels.
- Global names starting with *\$DELETE.\$PENDING* are reserved, and may not be used.

Software Object Platform Dependencies

A software object's format must be compatible with the method used at a target to install software. When builders create software objects, they must specify the type of installation procedure that will be used, which can be:

- Generic

- CID

The software installation methods supported by TME 10 Software Distribution depend on the type of workstation where the installation is being performed. All the methods described below require that software objects be stored at the server. A software object can, however, contain the specification of software files to be accessed remotely for the actual installation process. This applies whether the software installation process is initiated from a server or from an active client workstation.

Generic Software Objects

Generic software objects are created when the generic installation method is used. It is a robust all-purpose method that can be used to install on targets on all supported platforms. This method replicates the files contained in a software package.

CID Software Objects

The configuration, installation, and distribution (CID) method can be used to prepare software for clients on all supported platforms.

Use CID software objects for:

- IBM products, such as Database Server or Communications Server
- Other products that conform to the CID standards and that are extensively affected by the target workstation hardware and software configurations

The CID method of installation uses redirection of code images from files installed on the hard disk of any workstation within the same domain as the target workstation. The code images are not required to be stored on the server. Response files that allow for unattended installation are prepared at the preparation site.

The redirection of code images can be performed by the Network File System (NFS) or by the IBM File Server/Requester. The method used depends on the workstation from which the code images are to be redirected.

The TME 10 Software Distribution Catalog

The TME 10 Software Distribution catalog is a change control history database maintained on the server. The catalog contains a list of the names of all software objects and data files that are available to authorized users of client workstations across the network. It also lists the status of software objects at each workstation.

Items in catalogs are identified using labels so that each is unique across a multiserver, multidomain environment. Parts of the software object's label can also be used to identify the software platform a target runs under.

Items can be added to or removed from the catalog by any authorized user. The objects named in the catalog can reside on the server or on any of the clients in the domain.

Installing the Same Software Object on Different Workstations

The catalog can be used by all the clients associated with the server on a single-server network to check the availability of new software packages, updates to existing software packages, and data files that are available for general distribution to authorized client workstations.

Installing the Same Software Object on Different Workstations

You can create software objects that are *static*, meaning that the installations they trigger are always the same regardless of the workstations they are installed on, or *dynamic*, meaning that the installation procedures are different depending on the configuration of the workstations they are performed on.

Dynamic software objects are defined by expressing conditions that must be satisfied before a corresponding operation takes place. You can specify dynamic conditions in relation to:

- The hardware installed on a workstation
- The software installed on a workstation
- The operating system running on a workstation
- The directory the software is to be installed in, or already resides in if the software is being uninstalled or removed
- Procedures to be run at the workstation before or after operations are performed.

For example, you can take advantage of the flexibility offered by dynamic software objects to:

- Install software on two types of workstation: small machines or big machines. To do so you would include these conditions:

```
DYNAMIC SECTION: big_machine
CONDITION:          DRIVE_SIZE >= 180M
PRE-INSTALL:        <product_directory>\pre-install

DYNAMIC SECTION: small_machine
CONDITION:          DRIVE_SIZE < 180M
PRE-INSTALL:        <product_directory>\pre-install-small

OBJECT:
DYNAMIC SECTION:    small_machine
SOURCE NAME:        <product_directory>\file_small
TYPE:               FILE
ACTION:             COPY
```

Installing the Same Software Object on Different Workstations

- Mount different remote directories for installations on workstations with different operating systems: one for UNIX-based operating systems, one for Intel-based operating systems, and one for Windows NT. To do so you would include these dynamic sections:

```
GLOBAL NAME:                user4.exit4.ref.1
CHANGE FILE TYPE:           GEN
COMPRESSION TYPE:           LZW
REBOOT REQUIRED:             NO
DEFAULT TOKEN:              SRCINST(AIX) = /mnt/
DEFAULT TOKEN:              TRGINST(AIX) = /instago/
DEFAULT TOKEN:              TRGINST(ALL) = C:\INSTAGO\
DEFAULT TOKEN:              SRCINST(ALL) = z:\
REMOVABLE:                  YES
ACTIVABLE:                  YES
INTERACTIVE:                NO
AUTHORIZE:                  NONE
SW HISTORY RESET:          NO
INSTALLATION DURATION:      00:00:00
COST:                        0
PACK FILES:                 NO
SECURE PACKAGE:             NO
```

```
DYNAMIC SECTION:  intel_mount
CONDITION: OPS.OPERATING_SYSTEM=OS/2
REMOTE DIRECTORY:
  SERVER NAME:              os2server domtst
  EXPORTED DIRECTORY:       c:\usrdir
  MOUNTED FILE SYSTEM:      z:
  MOUNT OPTIONS:            alstst
```

```
DYNAMIC SECTION:  unix_mount
CONDITION: OPS.OPERATING_SYSTEM=AIX
REMOTE DIRECTORY:
  SERVER NAME:              pwpc1
  EXPORTED DIRECTORY:       /alfi/newbld/client_2
  MOUNTED FILE SYSTEM:      /mnt
PREREQ COMMAND:            ls -l / > /ls.out
```

```
DYNAMIC SECTION:  nt_mount
CONDITION: OPS.OPERATING_SYSTEM=WINDOWS_NT
REMOTE DIRECTORY:
  SERVER NAME:              650
  EXPORTED DIRECTORY:       source.nt
  MOUNTED FILE SYSTEM:      L:
  MOUNT OPTIONS:            /user:guest
PREREQ COMMAND:            dir D:\ > D:\dir.out
```

```
OBJECT:
  DYNAMIC SECTION:         intel_mount
  SOURCE NAME:             c:\usrdir
  SOURCE NAME AT INSTALL:  $(SRCINST)fname.txt
```

Software Preparation Interfaces

```
TARGET NAME:      $(TRGINST)fname.txt
TYPE:             REMOTE_FILE
ACTION:           COPY
INCLUDE SUBDIRS:  NO
```

```
OBJECT:
  DYNAMIC SECTION:  unix_mount
  SOURCE NAME:      /mnt1/client_2/fndswinv
  SOURCE NAME AT INSTALL: $(SRCINST)fndswinv
  TARGET NAME:      $(TRGINST)fndswinv
  TYPE:             REMOTE_FILE
  ACTION:           COPY
  INCLUDE SUBDIRS:  NO
```

```
OBJECT:
  DYNAMIC SECTION:  nt_mount
  SOURCE NAME:      /mnt1/client_2/fndswinv
  SOURCE NAME AT INSTALL: $(SRCINST)NTREF.BAT
  TARGET NAME:      $(TRGINST)NTREF.BAT
  TYPE:             REMOTE_FILE
  ACTION:           COPY
  INCLUDE SUBDIRS:  NO
```

Software Preparation Interfaces

You can use one of three methods to prepare software objects:

Graphical interface preparation notebook

The graphical interface allows you to create both basic software objects that simply contain software files and any scripts or procedures, and advanced software objects that include additional functions such as dynamic conditions and remote directory sources.

Graphical interface DiskCamera function

The DiskCamera function available in the graphical interface facilitates the creation of software objects by “taking pictures” of a hard disk before and after a software product is installed on it. The two pictures are compared by DiskCamera, which generates a software object that includes all new files the comparison found on the drive as well as any changes that were made to system files as a result of the installation.

Command line interface

You create software objects using the command line interface by first creating an ASCII software object profile and then issuing the build command against it.

Chapter 4. Using Change Control Operations

This chapter describes change control operations made available through TME 10 Software Distribution, and how to schedule their use in your network.

Change Control Operations

TME 10 Software Distribution offers you a fully automated method for distributing and installing software. You can use it to schedule and install software and updates to previously installed software, as well as to uninstall updates and restore the software to its previous state. TME 10 Software Distribution performs the following functions in a network:

- ***Installs software packages on workstations***

A new software package can be installed automatically, with no user intervention, on all workstations in the network. Installed software is available for use immediately after a client is restarted. This is done using the *install* function for a *refresh* software object.

Software changes can also be distributed over a network so that software files are either updated or removed automatically on all workstations on which the software was originally installed. An installation request can also check that prerequisite and corequisite software is present on a target before installing software packages.

- ***Applies updates to a current software level***

When a new level of a software product is released, it can be applied to existing software as an update.

- ***Applies fixes to software***

When fixes to installed software are released, these can be applied to all existing installed copies. This is done using the *install* function for a *fix* software object.

- ***Installs software from remote workstations***

Software files to be installed can be stored in a remote directory that is mounted at a target during the installation process. When you do not mount remote directories for installations, files are sent to a target, stored in the work area during installation, and then deleted when the installation is complete. This means that a target must have double the amount of disk space required by a product.

When you take advantage of the remote mount capabilities offered by TME 10 Software Distribution, you can redirect installations:

- To directories located on a server.

You do so by specifying the name of the directory to be exported and the options required for the remote mount using specific shared tokens (SERVERREPOS, SEVEREXPOP, REMOTEREPOS, REMREPMNTOP). You specify them when you define target parameters using the **addpm** command.

- To directories located on another workstation in the domain.

Scheduling Change Control Operations in the Network

You do so by specifying the directory to be exported and the remote mount information when you build a software object using either the graphical interface or the **bld** command.

- ***Installs software on unattended workstations***

You can use TME 10 Software Distribution to send a command to a client workstation to install, update, or delete changes to its software while it is unattended and has no user logged on.

- ***Rolls back (backs up) to the previous level of software***

An installation operation can also include a request to back up the version of software currently installed on a workstation. Any files deleted or modified by the installation process are automatically saved, and if the installation is unsuccessful, the previous version can be restored, or rolled back. The current level of the application must be *installed* with the *removable* option in order for the *remove* function to perform a roll-back.

- ***Makes changes permanent***

If the software package was installed on one or a group of clients with the backup option, a single *accept* request to make this version permanent can be submitted to all the clients. After software has been accepted, the previous version or level cannot be restored.

- ***Uninstalls one or more software packages installed previously by TME 10 Software Distribution***

The *uninstall* function removes, in a single operation, selected software packages from one or more clients. The software packages must have been installed originally by TME 10 Software Distribution.

- ***Activates software on one or a group of workstations***

The *activate* function restarts one or a group of workstations in a single operation after TME 10 Software Distribution has been used to install software packages on them.

Scheduling Change Control Operations in the Network

You can schedule the distribution of data and software and change control operations to occur at specific times, according to when it is both necessary and most convenient. Using appropriate schedules you can achieve network-wide synchronization of the installation of particular software. The scheduling facilities provided by TME 10 Software Distribution are flexible, and provide you with alternative methods to express when and how often operations should take place. You can:

- ***Specify change control windows for targets***

Each client workstation that is a target for software installation can be configured with a *change control time window*, a period of time during which installation of the software can take place. Installation of software that is scheduled for times outside the change control window for a client workstation is suspended until the change control window is open.

You specify change control windows when you define targets to the network.

- ***Specify origin or destination operation times***

You can schedule operation times for either the workstation where the operation originates, or the workstation where the operation is to take place. This makes it possible to schedule requests that are to take place in time zones different from the zone of the origin workstation. You can also schedule a period of time *before* and *after* which an operation cannot take place.

- ***Specify high-priority operations***

Change control operations that are urgent can be transmitted as high-priority, meaning that they will take place as soon as the currently executing operation has completed.

- ***Specify recursive operations***

A *recursive* operation is one that is performed repeatedly at specified intervals. Intervals can be specified on a monthly, weekly, daily, or hourly basis, depending on business requirements.

The execution of occurrences of recursive plans can be conditioned on the success of previous occurrences. This means that a subsequent transmission can take place only if the previous one produced the expected result. If Monday's sales data was not sent successfully, Tuesday's occurrence of the same operation will not be generated and transmitted.

- ***Hold operations at targets***

A user at a client workstation can temporarily suspend or prevent the execution of change control activities requested by the administrator by holding change control activity or by turning the workstation off. The functions are held and are executed when the workstation is switched on or the workstation is released.

Distributing Data Files

The term *data distribution* refers to the exchange of data files between client and server workstations within and across domains. You can use TME 10 Software Distribution facilities to distribute or retrieve data files such as:

- User data files
- Software packages and documentation
- Problem management data files such as dumps, log files, and trace files

TME 10 Software Distribution can distribute data files whether they have been cataloged or not. In addition, it provides facilities to compress and translate data during distribution.

Compressing Data Files

When large volumes of data are transmitted across a network, transmission efficiency is a significant factor in reducing time and cost.

Installing on Pristine Workstations

TME 10 Software Distribution can send or retrieve data in compressed form. The compression format used can be either SNA, LZW, or based on a user-supplied algorithm. Compression substantially reduces both line transmission time and storage space requirements. After transmission, data can be expanded into its original form.

Data compression facilities are available for data exchange within and across domains when your network uses the server-to-server communication protocol.

Translating Data

When different environments in your network use different data encoding techniques, TME 10 Software Distribution can perform data translation on files containing character data. A typical example of this requirement is transmitting data between systems that use ASCII encoding and those that use EBCDIC encoding.

Data translation facilities are available for data exchange within and across domains.

Data Distribution Functions

The distribution functions of TME 10 Software Distribution are:

- | | |
|-----------------|---|
| Send | The send function is used to send a file from a source target, which must be either the server or a local client workstation, to a destination target or group of targets, which can be either local or remote. The send function can be used from both server and client workstations. The action can be immediate or deferred until a specified time and date. It can also be issued with the replace option. |
| Retrieve | The retrieve function is used to retrieve a file from another target, which can be local or remote, onto a local server or client workstation. The retrieve function can be used from both server and client workstations. The action can be immediate or deferred until a specified time and date. |
| Delete | The delete function is used to delete files at local or remote targets. |

Installing on Pristine Workstations

TME 10 Software Distribution can be used to install software on pristine workstations (workstations that have no software installed). For examples, see the INF file *Pristine and Migration Scenarios* in the product Information folder.

Chapter 5. Tracking Network Operations

This chapter describes the facilities provided by TME 10 Software Distribution to track the hardware and software configuration of the workstations in your network.

Keeping Track of Workstation Hardware Configurations

You can use TME 10 Software Distribution facilities to keep track of the hardware configuration of the workstations in your network. This information is referred to as hardware inventory.

You run an inventory discovery procedure to detect what hardware is present on a workstation. The output of the procedure is stored in a file in the server catalog. You can then make use of this hardware inventory information when you formulate the conditions for creating dynamic groups and software objects, and for change control and distribution requests. For instance, you can specify that a software object be installed only on workstations that have at least 10 MB of disk space available. Or you can define a dynamic group that contains only workstations with a graphic adapter installed, and install a graphics program only on those workstations.

Inventory discovery is not a mandatory part of TME 10 Software Distribution. You can specify the hardware present on a target when you define it by specifying hardware parameters. However, the inventory discovery process is a way of catching this information and keeping it up to date automatically.

You can run inventory discovery on remote targets if the targets involved in the operation are connected by the server-to-server (STS) transmission protocol.

An inventory discovery program that takes advantage of NetFinity inventory capabilities on workstations that run OS/2 and Windows TME 10 Software Distribution products is provided. It is called `fndinv`, and it creates at each target output files (`fndswinv`, `fndhwinv`, and `fndtkinv`) that are then used by TME 10 Software Distribution.

Inventory data is transferred from clients to the server when you do either of the following:

- From the command line interface, submit the **`nvdms inv`** command.
- From the graphical interface, select **Inventory** from the **Selected** pull-down menu in the Targets window.

Note that for `fndinv` to work, the OS/2 or Windows client must have the Hw/Sw Discovery Tool component installed.

Keeping Track of Workstation Software Configurations

You can keep track of the software installed on the workstations in your network using two methods:

Keeping Track of Workstation Software Configurations

- Running software inventory procedures
- Referring to the change control status information in the catalog.

Software Discovery Information

You can run inventory discovery procedures to detect the software installed on workstations. These procedures are similar to those described for hardware inventory in “Keeping Track of Workstation Hardware Configurations” on page 35.

Change Control Status Information

Change control status is stored in the server catalog for each software package installed on each client workstation in a domain. The catalog also includes the change control history for the software installed on the server itself. When a change control operation is executed on a client workstation, a report of the operation results is generated and routed to the workstation in the network that originated the operation.

The status information in the catalog can report a software object as being:

Active/Inactive: The software object is installed either in the active area of the target file storage or only in the service area.

Available/Not authorized: The software object is stored on the server and is available for use by the target, or the target has not been authorized to use it.

Back level: The software object is a previous version of the software that was backed up after a more recent version was installed, but is reactivated because the more recent version was removed.

Discovered: The software object was discovered by an inventory discovery procedure. It is installed on the workstation, but was not installed by a TME 10 Software Distribution procedure.

Distributed: The software object was distributed to this target from another target.

Distribution pending: The software object is currently being distributed.

In error: A change control operation using the software object has failed without recovery, leaving the installation in an unpredictable state.

Installed/Not installed/In progress: The software object is either installed, is not installed, or the installation is currently in progress.

Reboot required: When the software object was installed, REBOOT REQUIRED=YES was specified. To use the new software, restart the workstation so that the changes in the file become operative.

Removable/Not removable: The software object is installed on the target and can be removed and replaced with an earlier version (rolled back), or cannot be removed.

Before sending software objects to clients, you can check the status of the software on the client. The operations that you can schedule for a client depend on the status of software, as shown in Table 6. You can use change control status information to specify conditions in dynamic software objects and dynamic groups (see “How Grouping Targets Can Facilitate Change Control Operations” on page 23 and “Installing the Same Software Object on Different Workstations” on page 28). For example, you can schedule a remove operation only on those workstations in your network where the status of a software object is *in error*.

Table 6. Software Object Status and Associated Functions

Status	Install	Accept	Activate	Remove	Uninstall
Available	Yes	No	No	No	No
Distributed	Yes	No	No	No	No
In error	Yes	Yes	Yes	Yes	Yes
In progress	No	No	No	No	No
Installed, Removable, Active	No	Yes	No	Yes	Yes
Installed, Not removable, Active	No	No	No	No	Yes
Installed, Removable, Inactive	No	Yes	Yes	Yes	Yes
Installed, Not removable, Inactive	No	No	Yes	No	Yes

Tracking the Progress of Change Control Operations

Change control and distribution operations that have been submitted to the system for execution are referred to as *requests*. You can display requests to follow their progress and take any necessary remedial action. Detailed filtering mechanisms allow you to display only those requests you are immediately interested in. You can view the progress of requests according to:

- The servers they were submitted from
- The request queue sequence number
- The global name of the files included in them
- The type of operation being performed (install, remove, accept, activate)
- Their status (in progress, successful, waiting, held, deleted, failed, not started)
- The severity level returned for errors (failed, warning, severe, hardware failure)
- The scheduled date and time
- The domains or the targets they address

You can then act upon a request by:

- Holding or releasing its execution
- Deleting it
- Rescheduling it
- Restarting it

Chapter 6. Using Security Functions

TME 10 Software Distribution provides security mechanisms that you can use to prevent unauthorized use of product functions and safeguard the privacy of sensitive data. You can define:

- User profiles that limit the functions a user can perform
- User access to targets
- User access to data
- Target authorizations for specific software objects
- Security checks to be performed on the files in software objects before they are installed on workstations.

TME 10 Software Distribution users must have a user ID to access the product. You can provide additional security by assigning passwords to users as well.

The following sections describe the authorizations contained in user profiles, and how to define user access to data and to targets.

Defining User Authorization Profiles

User authorization profiles define the functions that each user can perform with the product. You can define profiles to be used by groups of users who perform similar tasks, or you can define individual profiles for each user.

You can take advantage of three default authorization profiles, which are created during installation and stored at the server.

Administrators (FNDADMN)

Administrators have access to *all* operations, including the administrative and configuration functions.

When you define user authorizations, keep in mind that all FNDADMN users can perform operating system specific operations that require root and bin authorizations.

Table 7 on page 40 lists the values set for this profile.

Defining User Authorization Profiles

<i>Table 7. Administrator Profile (FNDADMN)</i>	
Function	Authorization
Change management	Install, Remove, Accept, Uninstall Execute Activate Authorize, Unauthorize, Delete History All targets
Distribution	Send Retrieve, Delete
Preparation	Build, Unbuild, Catalog, Delete, Create, View
Queues	Manage
System administration	Modify
Configuration	Modify
Erase requests	Authorize
Manage requests	All

Builders (FNDBLD)

Builders are authorized to perform change control preparation functions. That is, they can prepare and build software objects. Table 8 lists the values set for this profile.

<i>Table 8. Builder Profile (FNDBLD)</i>	
Function	Authorization
Change management	Install, Remove, Accept, Uninstall Execute Activate All targets
Distribution	Send Retrieve, Delete
Preparation	Build, Unbuild, Catalog, Delete, Create, View
Queues	View
System administration	View
Configuration	View
Erase requests	Authorize
Manage requests	All

Users (FNDUSER)

Users can distribute files and display the configuration. Table 9 lists the values set for this profile.

Table 9. User Profile (FNDUSER)	
Function	Authorization
Change management	Install, Remove, Accept, Uninstall Execute Activate Only on target where the user is working
Distribution	None
Preparation	None
Queues	View
System administration	None
Configuration	None
Erase requests	No authorization
Manage requests	None

Defining User Access to Data and Targets

Define data access keys (DAKs) to determine the objects in the catalog that a user can work with, and define target access keys (TAKs) to determine the targets a user can access. You define them in the following way:

- When you build or catalog a software object, a data file, or a plan, you specify the DAK associated with it. A catalog entry can have only one DAK, but you do not have to assign it one at all if it is not necessary. No DAK is the default.

In the same way, when you define a target you specify the TAK associated with it.

- When you define a user, you determine the catalog entries the user can access by specifying the DAKs the user is associated with, and the targets the user can access by specifying the user's TAKs. A user can have up to 32 DAKs and TAKs (which is the maximum number defined to the system). The default is no DAKs or TAKs, in which case the user can work only with objects or targets that do not have DAKs or TAKs.

When a user issues a command against a catalog entry or for a target, the system checks to verify that the object or target and the user have the same DAK or TAK.

Defining Data Security

Since software objects are usually prepared at preparation site workstations, then stored on servers before being distributed around a network and installed, it is important to ensure that their contents are not tampered with before an actual installation takes place. This is especially critical when a software object contains sensitive data.

TME 10 Software Distribution provides the following mechanisms:

- When you are building a remote source software object, specify **Secure package** in order to:
 - Verify that the remote files specified are accessible, when the software object is built.
 - Verify that the files are *the same* as those specified when the software object was built, when the software object is installed.
- In addition, you can activate user exits that perform the following functions in connection with the secure package attribute:
 - When the software object is built, assign a *secure key* to it by calculating its CRC (Cyclic Redundancy Check) number.
 - When the software object is installed at a client, the client product proceeds with the installation only if the secure key is the same as the one originally assigned, and only if the secure key identifies the workstation where the software object was built.

Chapter 7. Finding and Using TME 10 Software Distribution Information

This chapter tells you where you can find more information about TME 10 Software Distribution functions and how to use them.

It makes reference to files that can be found on the media on which the product is provided. TME 10 Software Distribution is provided on two CD-ROMs:

CD Number	CD Name	Product Number
LK3T-5087-00	TME 10 Software Distribution, Version 3.1.5 for OS/2, Windows 2000, Windows NT, Windows 95/98, Windows 3.1x, NetWare	5698-SWD
LK3T-5088-00	TME 10 Software Distribution, Version 3.1.5 for AIX	5698-SWD

Information about Messages and Error Codes

All TME 10 Software Distribution messages are logged in the message log. You can view it in the Message Log window of the graphical interface, or by entering the `nvdmm log` command from a command prompt.

Viewing Online Information

If you installed the Distribution Server Documentation or Distribution Client Documentation option, then to see files of online information, click on the Documentation icon in the product folder. The Documentation folder, shown in Figure 7, is displayed.

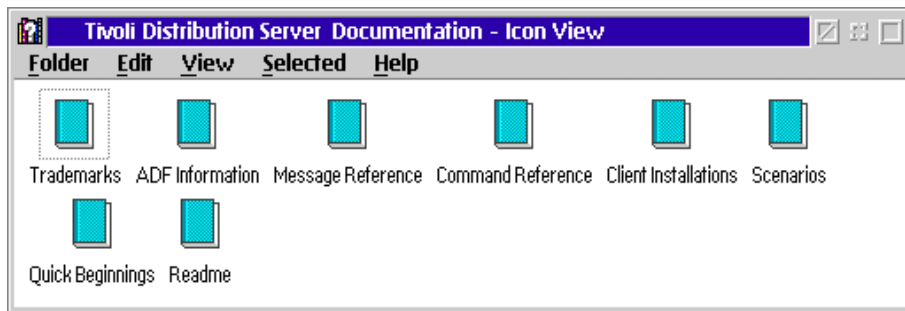


Figure 7. Distribution Server documentation - icon view

Viewing Online Help

The icons in the folder represent these items:

- Quick Beginnings, an INF file (FNDOSMST.INF) of this book.
- Clients Installation and Configuration, an INF file (FNDCIMST.INF) of the manual that explains how to install and configure clients for the OS/2, AIX, Windows 2000, Windows NT, Windows 98, Windows 95, and Windows 3.11 platforms.
- A Readme file (README.INF), which contains information about limitations and about changes made to TME 10 Software Distribution, Version 3.1.5 for OS/2 or to the publications after the publications went to press.
- Command Reference, an INF file (FNDR6MST.INF) that contains reference information about the TME 10 Software Distribution, Version 3.1.5 for OS/2 command line interface.
- Message Reference, an INF file (FNDM6MST) that contains reference information about the messages issued by TME 10 Software Distribution, Version 3.1.5 for OS/2.
- Pristine and Migration Scenarios, an INF file (FNDS6MST.INF) that contains detailed examples of the use of TME 10 Software Distribution to install software on pristine workstations and to migrate from one version of an operating system to another.
- ADF Information, an INF file (ADF.INF) that explains how to create the file that is required to prepare an application for CID installation through TME 10 Software Distribution.
- Trademarks, an INF file (TRADEM.INF) that lists trademarks used in the product.

To display an INF file, click on the icon or use the VIEW command.

Viewing Online Help

Help information is available for every menu item on pop-up and pull-down menus, and every place you see a **Help** push button.

General Help

F1 help provides information about objects, pop-up menu items, entry fields, and push buttons. To use the F1 help, select or highlight an item and press the F1 key on your keyboard.

Another way to get help is to use the **Help** push button available on most notebook pages, and the **Help** menu item on most object pop-up menus. These sources enable you to work with an object while you learn how to use it.

Message Help

Messages issued for selected TME 10 Software Distribution, Version 3.1.5 for OS/2 services are displayed in a pop-up window that has a **Help** push button. Select this push button to see an explanation of the message.

From the command line, you can get message help by issuing the command:

```
HELP FNDM6MST <msgno>
```

where *msgno* is the message number.

Command Help

To obtain help with the syntax of a software distribution command, do the following:

1. At the OS/2 command prompt type:

```
NVDM
```

The following prompt appears:

```
NVDM>
```

2. At the NVDM> prompt type:

```
HELP <command> -
```

where:

```
<command>
```

is the command name.

The syntax and parameters for the command appear; for example:

```
NVDM> HELP LSTG
```

gives you the syntax and parameters for the command LSTG.

From the command line, you can get command help by issuing the command:

```
HELP FNDR6MST <cmdname>
```

where *cmdname* is the name of the command.

Printing the Documentation

TME 10 Software Distribution documentation is available on the product CD in PostScript and PDF formats.

PostScript Format

Files of the TME 10 Software Distribution documentation formatted for a PostScript printer are supplied on the product CD-ROM under the directory SD4DOCPS, which contains the following files:

Calling IBM Service

Filename	Document name
FNDNTMST.PS	TME 10 Software Distribution for Windows NT Quick Beginnings
FNDOSMST.PS	TME 10 Software Distribution for OS/2 Quick Beginnings
FNDNWMST.PS	TME 10 Software Distribution for NetWare Quick Beginnings
FNDAXMST.PS	TME 10 Software Distribution for AIX Quick Beginnings
FNDR6MST.PS	TME 10 Software Distribution for Windows NT and OS/2 Command Reference
FNDA6MST.PS	TME 10 Software Distribution for AIX Reference
FNDNRMST.PS	TME 10 Software Distribution for NetWare Command Reference
FNDCIMST.PS	TME 10 Software Distribution Clients Installation and Configuration
FNDM6MST.PS	TME 10 Software Distribution Message Reference
FNDS6MST.PS	TME 10 Software Distribution Pristine and Migration Scenarios
FNDI6MST.PS	TME 10 Software Distribution for AIX Installation Scenarios
README.PS	TME 10 Software Distribution README

You can print them before or after you install TME 10 Software Distribution, Version 3.1.5 for OS/2, using whatever method you have set up for printing to a PostScript printer. Because the files are large, use of a high-speed PostScript printer is recommended.

PDF Format

The TME 10 Software Distribution manuals are also available in PDF format, allowing them to be viewed or printed using Adobe Acrobat Reader, which can be downloaded free of charge from Adobe's Internet site (www.adobe.com); see the site for full details of platforms supported by Acrobat. The directory which holds the PostScript files also holds the same manuals in PDF format, where the filename of the manual is the same as that of the PostScript version but the extension is changed to .PDF.

The manuals are fully hyperlinked, so that you link directly to pages from the Table of Contents and the Index, as well as from specific references in the text.

Calling IBM Service

The IBM Support Center provides telephone assistance in problem diagnosis and resolution in the United States and Puerto Rico. You can call the IBM Support Center at any time; you will receive a return call within eight business hours (Monday through Friday, 8 a.m. to 5 p.m., your local time). The number to call is (800) 237-5511.

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

Part 2. Installing and Configuring TME 10 Software Distribution

This part describes what you must do to install the product and configure the network it will be running in. It includes these chapters:

- Chapter 8, "Planning" on page 49
- Chapter 9, "Installing an OS/2 Distribution Server" on page 53
- Chapter 10, "Editing the Base Configuration File" on page 73Chapter 11, "Defining Users and Targets" on page 83
- Chapter 12, "Configuring for Communication with MVS" on page 91
- Chapter 13, "Configuring Personal Communications/Communications Server" on page 113
- Chapter 14, "Configuring STS Remote Connection Files" on page 123
- Chapter 15, "Configuring SNA/DS Remote Connection Files" on page 133

Information about how to install and configure OS/2 and Windows clients is in the *Installation and Configuration* book for the client products.

Chapter 8. Planning

This chapter contains the hardware and software prerequisites for installing the TME 10 Software Distribution server. You select the components you want to install. Your options are:

- Distribution server—required. The TME 10 Software Distribution catalog and the command line interface.
- Distribution server GUI—optional.
- Preparation site server—optional. The programs that support preparation of software objects.
- Hardware/software discovery tool—optional. The programs that support hardware and software inventory discovery. These programs come from the NetFinity product. If your workstation already has NetFinity Manager or NetFinity Services installed, do not install this component of TME 10 Software Distribution.
- Distribution server documentation—optional. The TME 10 Software Distribution manuals, message documentation, command documentation, and other information in INF format.

Hardware and software prerequisites for TME 10 Software Distribution clients are in the *TME 10 Software Distribution Clients Installation and Configuration* manual.

Hardware Prerequisites

- The processor and memory requirements of the TME 10 Software Distribution server are those required to run the Operating System on which it is to be installed; thus you should consult the Operating System documentation.
- Disk space for the TME 10 Software Distribution server, including the documentation, is approximately 45 MB
- 10 MB disk space required temporarily during installation
- Enough disk space to store the software objects for distribution.
- CD-ROM drive for product installation
- Token-ring or Ethernet card.

Software Prerequisites

The following is required on the distribution server workstation:

- From OS/2 Warp 3.0x to OS/2 4.5 Warp server for e-business
- Products to support the appropriate communication protocols (see “Communication Protocols” on page 50).
- To install CID-enabled products on clients, one of the following is required:
 - LAN Server and LAN Requester

Communication Protocols

- NFS
- SRVIFS
- NetWare Server and NetWare Requester

Communication Protocols

Communications protocols are required for server and client communication and for connecting an OS/2 distribution server to an MVS or AIX focal point.

Protocols to Connect to Another Server

The following protocols are supported for an OS/2 Server communicating with another TME 10 Software Distribution Server.

Server-to-Server (STS) Protocols

Table 10. STS Protocols to Connect to Another Server

Platform	Communications Protocol
OS/2	APPC IPX NetBIOS TCP/IP
NetWare	IPX TCP/IP
Windows NT or Windows 2000	IPX NetBIOS TCP/IP
AIX	APPC TCP/IP

SNA/DS Protocols: APPC is the only protocol available to connect to servers using SNA/DS.

Protocols to Connect to a Client

The following protocols are supported for an OS/2 server communicating with a TME 10 Software Distribution Client.

Table 11. Protocols to Connect to A Client

Platform	Communications Protocol
OS/2	APPC IPX NetBIOS TCP/IP
NetWare	IPX TCP/IP
Windows NT or Windows 2000	IPX NetBIOS TCP/IP
Windows 95 or Windows 98	NetBIOS TCP/IP
Windows 3.11	NetBIOS TCP/IP
AIX	TCP/IP

Protocols to connect to a focal point

For communication with an MVS focal point the APPC protocol should be used.

Server/Client Communications

The communications software requirements depend on the level of the operating system on which TME 10 Software Distribution is running:

Warp 3.0x to Warp 4.0

NetBIOS	Multiprotocol Transport Services (MPTS) 1.0 or later (LAPS level WR08000)	
	NetBIOS resources required on an OS/2 distribution server are 25 sessions, 25 commands, and 2 names.	
TCP/IP	IBM TCP/IP for OS/2; the version required depends on the level of MPTS/LAPS installed:	
	MPTS/LAPS level	TCP/IP Version to use
	WR08210	3.x
	WR08415	4.0 (see Info APAR II10529 for more details)
	WR08600	4.1
IPX/SPX	Novell NetWare Requester 2.10 or later	
APPC	Communications Server 4.1 or later or Personal Communications 4.1 or later.	

OS/2 4.5 Warp Server for e-business

With OS/2 4.5, the following resources are automatically installed as part of the Operating System package:

Prerequisites on the Host System

NetBIOS Multiprotocol Transport Services (MPTS) 5.5
NetBIOS resources required on an OS/2 distribution server are 25 sessions, 25 commands, and 2 names.

TCP/IP IBM TCP/IP for OS/2 4.2.1

IPX/SPX Novell NetWare Requester 2.10 or later

The following needs to be installed:

APPC Communications Server 5.0 or later.

Communications with a Focal Point

Communication with NetView DM for MVS will be via APPC; please see the APPC details given above.

For communication with TME 10 Software Distribution for AIX over TCP/IP, IBM TCP/IP for OS/2 3.0 or later, is required on the distribution server.

For communication with TME 10 Software Distribution for AIX over APPC, please see the APPC details given above.

Prerequisites on the Host System

To perform software distribution from an MVS system to TME 10 Software Distribution, the MVS system must have NetView DM for MVS Release 6.2 or 7 installed; however, if you wish to enable the host to delete pending requests at the server, it must be Release 7.

To perform TME 10 Software Distribution from an AIX system to TME 10 Software Distribution, the AIX system must have TME 10 Software Distribution 3.1.4 for AIX, or later, installed.

Chapter 9. Installing an OS/2 Distribution Server

This chapter explains how to:

- Install an OS/2 distribution server interactively from the CD-ROM (“Installing an OS/2 Distribution Server Interactively”).
- Use the APPC configuration SmartGuides to configure for communication with NetView DM for MVS, TME 10 Software Distribution for AIX, other OS/2 servers, and OS/2 clients (“Using the APPC Configuration SmartGuides to Connect Over SNA” on page 66).
- Configure for TCP/IP communication with TME 10 Software Distribution for AIX, both at the TME 10 Software Distribution site (“Configuring TCP/IP for Communication with a Remote Server” on page 68) and at the TME 10 Software Distribution for AIX site (“Configuring at the Remote Server Site” on page 69).

Installing an OS/2 Distribution Server Interactively

This section describes how to install a distribution server interactively from the CD-ROM.

Follow these steps:

- 1 Insert the TME 10 Software Distribution CD-ROM (number LK3T-5087-00) and, from the SD4OS2\IMAGES directory, type `install`. The Welcome window, shown in Figure 8 on page 54, is displayed.

Installing a Server

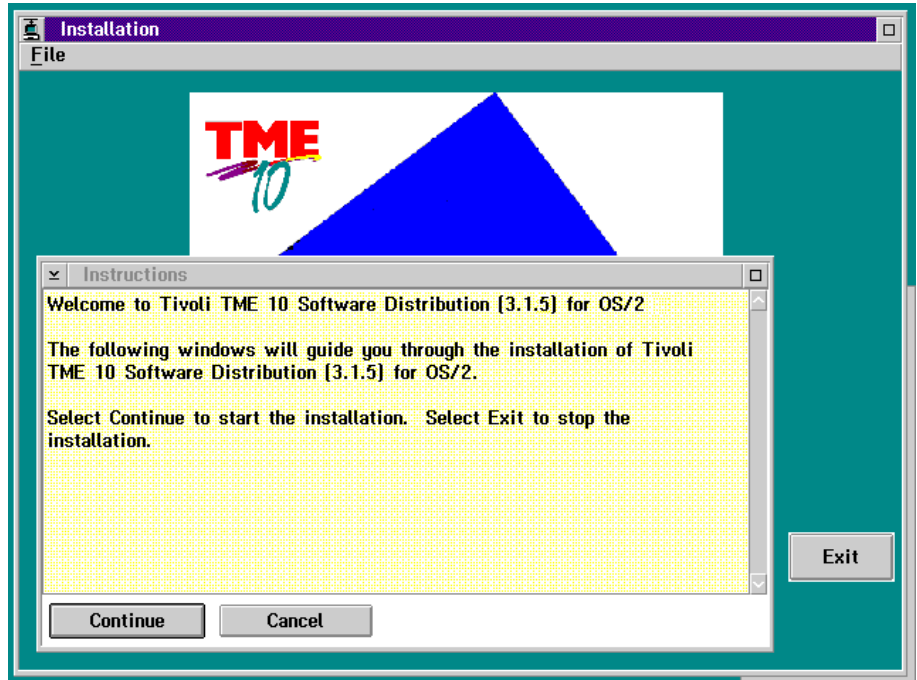


Figure 8. TME 10 Software Distribution welcome window

- 2 Select **Continue**. The Install window, shown in Figure 9, is displayed.

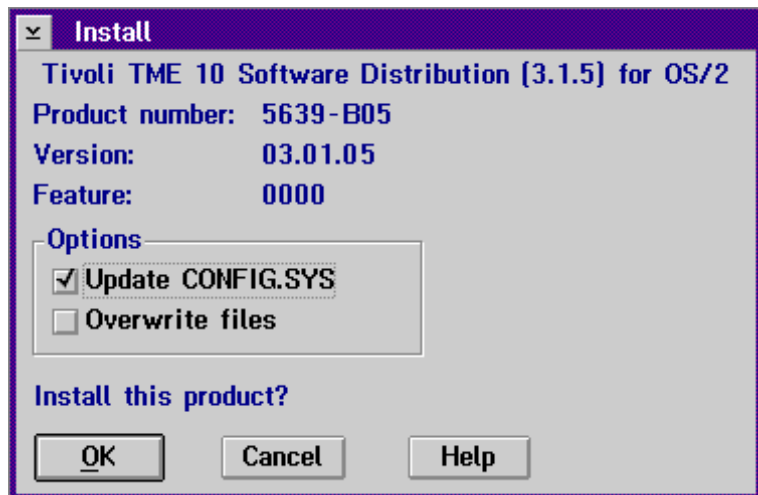


Figure 9. TME 10 Software Distribution install window

- 3 Select **Update CONFIG.SYS** and then select **OK**.

- 4 In the Install - directories window, shown in Figure 10 on page 55, select the components to install and the installation directory. This window includes the default for the TME 10 Software Distribution installation directory, SOFTDIST.

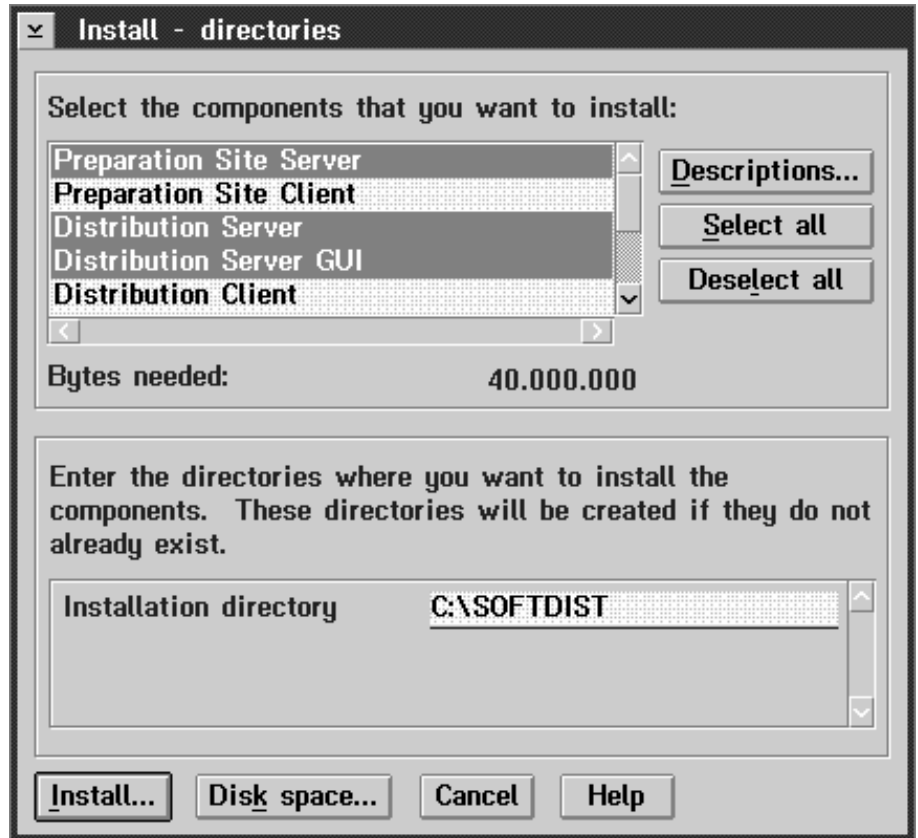


Figure 10. TME 10 Software Distribution install - directories window

Select the components you want to install and the drive on which to install the product.

It is recommended that you install Distribution Server Documentation; the online helps sometimes refer to the INF documentation, and if you have not installed these files, you will not be able to link to them from the helps. To view these files after installation requires no software other than the operating system.

- 5 Optionally, click on **Disk space...** to view disk space available on all your drives, as shown in Figure 11 on page 56.

Installing a Server

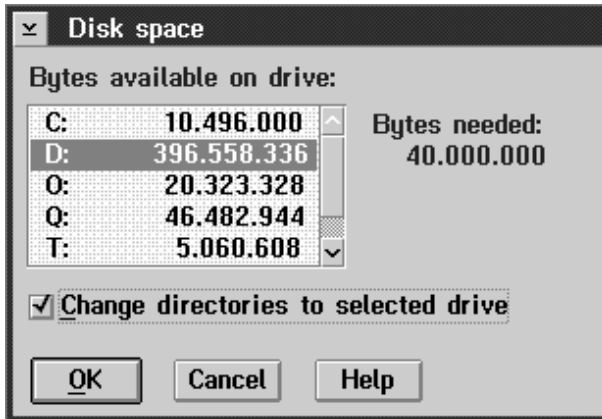


Figure 11. Disk space window

Select a drive and click on **OK**.

- 6 In the Install - directories window, select **Install**.

Installation begins. Progress is shown in the Install - Progress window, shown in Figure 12.

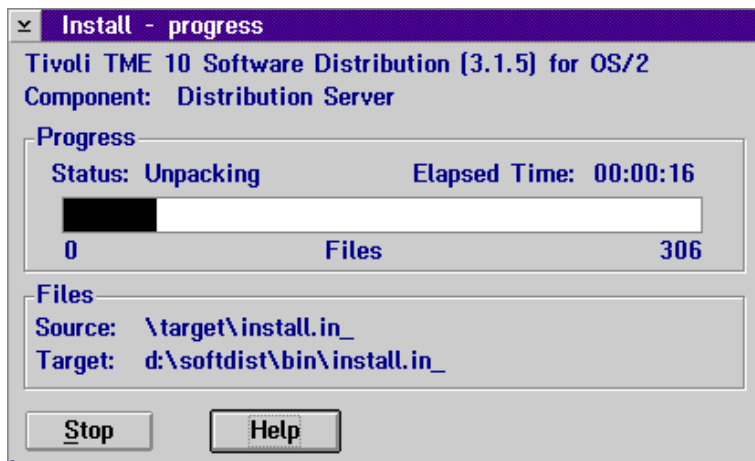


Figure 12. TME 10 Software Distribution install - progress window

- 7 Installation continues until the Distribution Configuration notebook, shown in Figure 13 on page 57, is displayed.

(Note that you can change the settings in this notebook after installation by using the Configuration icon in the TME 10 Software Distribution folder.)

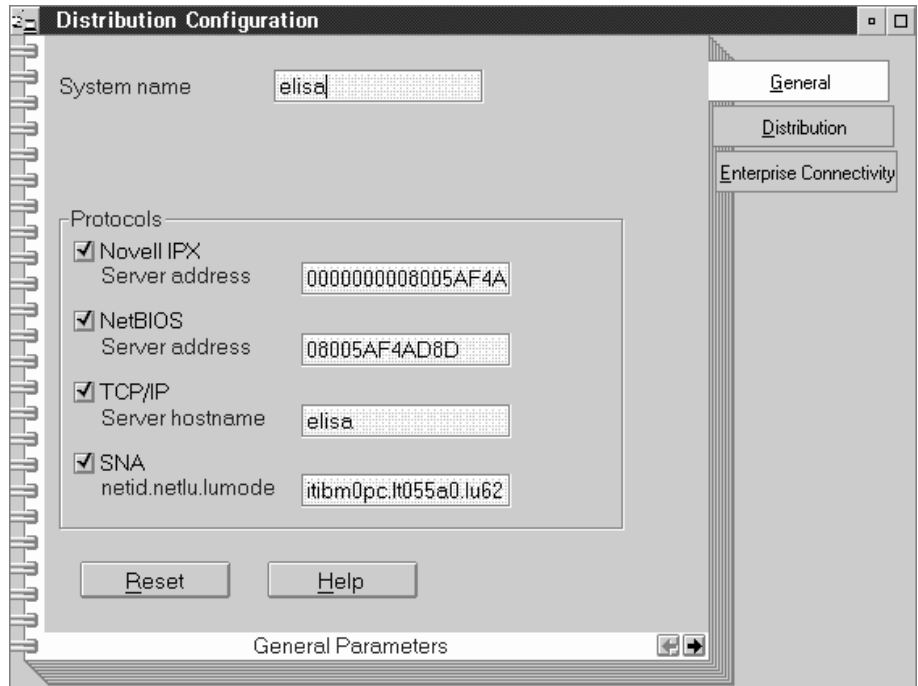


Figure 13. Distribution Configuration Notebook general page

Use the General page to specify the system name and the communication protocols to be used.

You can implement any combination of NetBIOS, IPX/SPX, TCP/IP, and SNA for communication between the distribution server and its clients. You can also implement SNA for communication with NetView DM for MVS. For example, if you configure NetBIOS and TCP/IP on the distribution server, the server can communicate with TME 10 Software Distribution clients that have either NetBIOS or TCP/IP, or both, configured.

Enter information in the following fields:

System name

Name of this system in the TME 10 Software Distribution environment. This name can be, but is not required to be, the same as the NetBIOS name or TCP/IP hostname. The name cannot contain embedded blanks, and cannot contain the characters * (asterisk), \ (backslash), and ? (question mark).

Note that the system name is case sensitive.

Protocols

Check the box next to the protocol name to enable the distribution server to communicate over the selected transport protocol. You can enable one protocol or multiple protocols.

Installing a Server

If you choose to enable SNA, fill in the values for *netid.netlu.lumode*. For the other protocols, the installation process fills in the required values.

- 8 Use the Distribution page of the Distribution Configuration notebook, shown in Figure 14, to select software distribution configuration parameters.

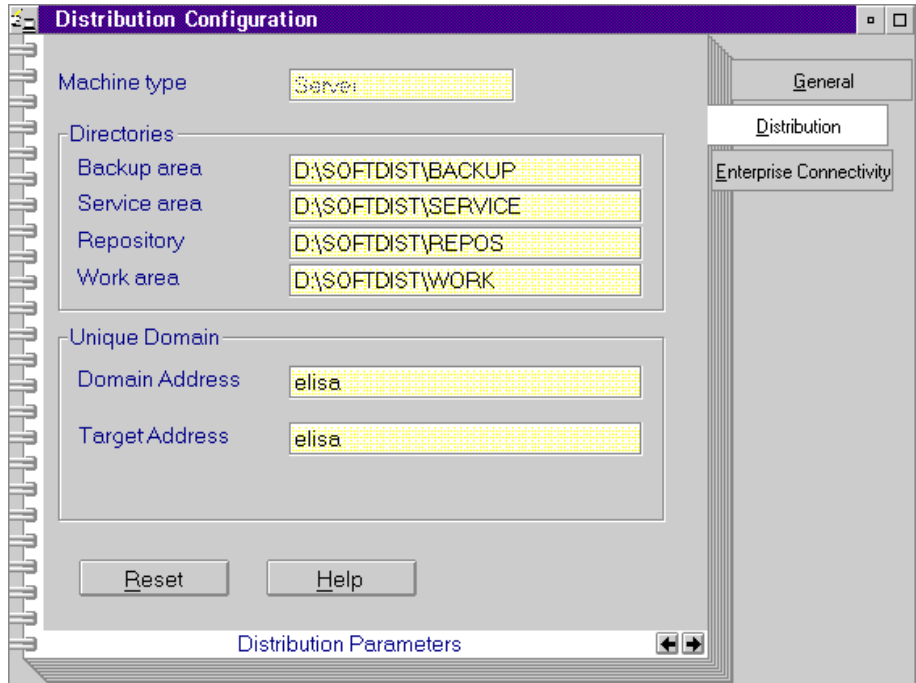


Figure 14. Distribution Configuration Notebook distribution page - Server

The default directories are shown in the figure.

The fields of this page are as follows:

Machine type

Software distribution workstation role. This entry is set to Server and cannot be changed.

Backup area

An area where backups of previous levels of applications are kept, for applications installed removably. For example, if you install Microsoft Word for Windows 2.0b on this machine and after six months you install Word for Windows 6.0, a backup of the previous version is stored in the backup area. This entry applies to applications installed through software distribution on *this* machine.

Service area

Temporary area for installations.

Repository

Subdirectory for profiles and software objects.

Work area Temporary area used by TME 10 Software Distribution during the resolution of change management requests.

Note: The paths that you specify in the Directories field group of the Distribution Configuration window must be the same path that you specify in the Installation Directory field of the Install - Directories window. If you want to change this path as in this scenario, edit the nvdm.cfg file after you complete the installation.

Domain Address

If you will be connecting to a distribution server on MVS or AIX, this is the name of the domain to which this TME 10 Software Distribution server and its distribution clients belong.

By default, this name is equal to the first eight characters of the **System name** you entered in the General page of this notebook.

Target Address

The name by which this distribution server is to be known by the MVS or AIX TME 10 Software Distribution server and the distribution clients in the LAN.

By default, this name is equal to the first eight characters of the **System name** you entered in the General page of this notebook.

- 9 If you plan to connect to NetView DM for MVS or TME 10 Software Distribution for AIX, use the last page of the Distribution Configuration notebook to select parameters related to enterprise connectivity, as shown in Figure 15 on page 60.

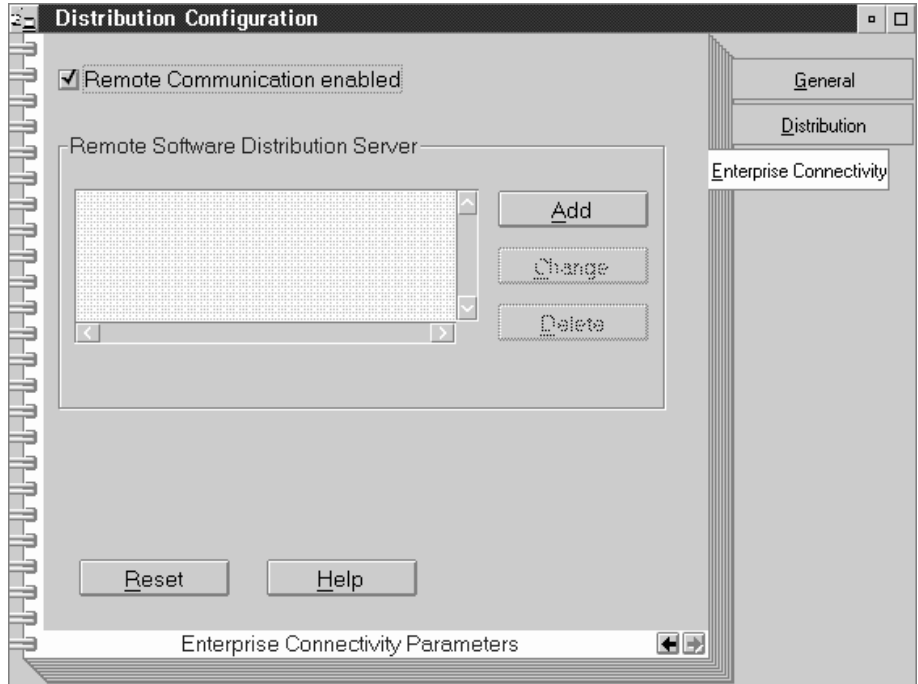


Figure 15. TME 10 Software Distribution Enterprise connectivity parameters

Enter information in the following fields:

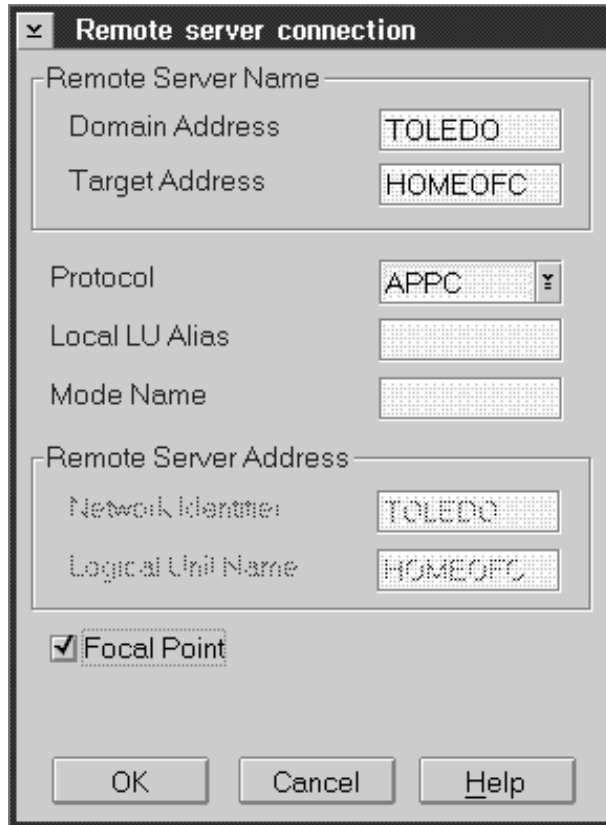
Remote Communication Enabled

Remove the check from this box if you want to disable remote communication with MVS and AIX.

Remote Software Distribution Server

Use this container to list all remote software distribution servers to which this distribution server can be connected. Each remote software distribution server is listed by domain name and server name. It can be either a NetView DM for MVS system or a TME 10 Software Distribution for AIX server.

When you click on **Add**, the Remote server connection window is displayed, as shown in Figure 16 on page 61.



The image shows a graphical user interface window titled "Remote server connection". It contains several input fields and a checkbox. The "Remote Server Name" section has "Domain Address" set to "TOLEDO" and "Target Address" set to "HOME OFC". The "Protocol" is set to "APPC" with a dropdown arrow. "Local LU Alias" and "Mode Name" are empty. The "Remote Server Address" section has "Network Identifier" set to "TOLEDO" and "Logical Unit Name" set to "HOME OFC". A checkbox labeled "Focal Point" is checked. At the bottom are "OK", "Cancel", and "Help" buttons.

Remote server connection	
Remote Server Name	
Domain Address	TOLEDO
Target Address	HOME OFC
Protocol	APPC
Local LU Alias	
Mode Name	
Remote Server Address	
Network Identifier	TOLEDO
Logical Unit Name	HOME OFC
<input checked="" type="checkbox"/> Focal Point	
OK	Cancel Help

Figure 16. Remote server connection window

Enter values in the following fields:

Domain Address

Enter the name of the network to which the remote server belongs. If the remote server is a NetView DM for MVS system, this name must be the same as the network ID of the partner LU in the Personal Communications APPC configuration of this workstation. If the remote server is a TME 10 Software Distribution for AIX system, this is the domain address of the TME 10 Software Distribution for AIX server.

The name can be up to eight characters long. The characters can be letters, numbers, or the special characters @, \$, and #.

Installing a Server

This name and the target address provide full identification for the remote server in the Remote Software Distribution Server container in the form:

<Domain address>.<Target address>

You can enter an asterisk (*) as a wildcard name. An asterisk as domain address implies that any of the defined domains is considered valid.

Target Address

Enter the remote server name.

If the remote server is a NetView DM for MVS system, the target address is the logical unit (LU) name assigned to NetView DM for MVS; it must therefore correspond to the name of the partner LU in the Personal Communications APPC configuration of this workstation.

If the remote server is a TME 10 Software Distribution for AIX server, this name is the target address of the TME 10 Software Distribution for AIX server.

The name can be up to eight characters long. The characters can be letters, numbers, or the special characters @, \$, and #.

This name and the Domain Address provide full identification, for the remote server in the Remote Software Distribution Server container, in the form:

<Domain address>.<Target address>

You can enter an asterisk (*) as a wildcard name. An asterisk as Target Address implies that any of the available remote servers is considered valid.

Protocol

Select the protocol that the remote server and this distribution server will use to communicate.

Select **APPC** if the remote server is a NetView DM for MVS system. This requires that Personal Communications be installed and configured on the TME 10 Software Distribution distribution server. Select **TCP/IP** if the remote server is a TME 10 Software Distribution for AIX server. This requires that TCP/IP be installed and configured on the TME 10 Software Distribution distribution server.

Some of the information requested changes according to which communication protocol you select here.

Local LU Alias (APPC only)

The LU alias name that identifies your software distribution program to Personal Communications when an APPC session is established with the NetView DM for MVS remote system.

Enter the local LU alias name that identifies this workstation, as defined in your Personal Communications APPC configuration.

The name is case sensitive, so it must be entered exactly as in the Personal Communications configuration.

If you do not enter a name here, Personal Communications uses the defined local LU name or local CP name in lieu of the local LU alias (if you have a working Personal Communications APPC configuration).

Mode Name (APPC only)

The name of the logon mode table entry, in the MVS system, containing the session parameters used for establishing the APPC session between NetView DM for MVS and this TME 10 Software Distribution distribution server. Enter the name of the mode as defined in your Personal Communications APPC configuration.

If you do not specify a mode here or in Personal Communications, then Personal Communications uses one of the default IBM-supplied modes. Note, however, that to communicate with the MVS remote system, the mode definition must include:

```
PLU_MODE_SESSION_LIMIT=1
```

Network Identifier (APPC only)

The name of the network to which the NetView DM for MVS system belongs.

If you already entered a domain address in this window, this field appears as read-only. If you did not enter a name in the **Domain Address** field, then you must enter one here.

Logical Unit Name (APPC only)

The logical unit name of the NetView DM for MVS remote server.

If you already entered a Target Address name in this window, this field appears as read-only. If you did not enter a name in the **Target Address** field, then you must enter one here.

HostName (TCP/IP only)

The host name of the remote TME 10 Software Distribution server.

Focal Point

Select this check box if the remote server is configured to function as a software distribution focal point.

This means that, beside being able to initiate software distribution operations on remote targets, the remote server can receive and store software distribution reports from this TME 10 Software Distribution distribution server.

Installing a Server

This box is grayed out if you already have a focal point defined.
Click on **OK** to add the remote server to the Remote Software Distribution Server container, as shown in Figure 17.

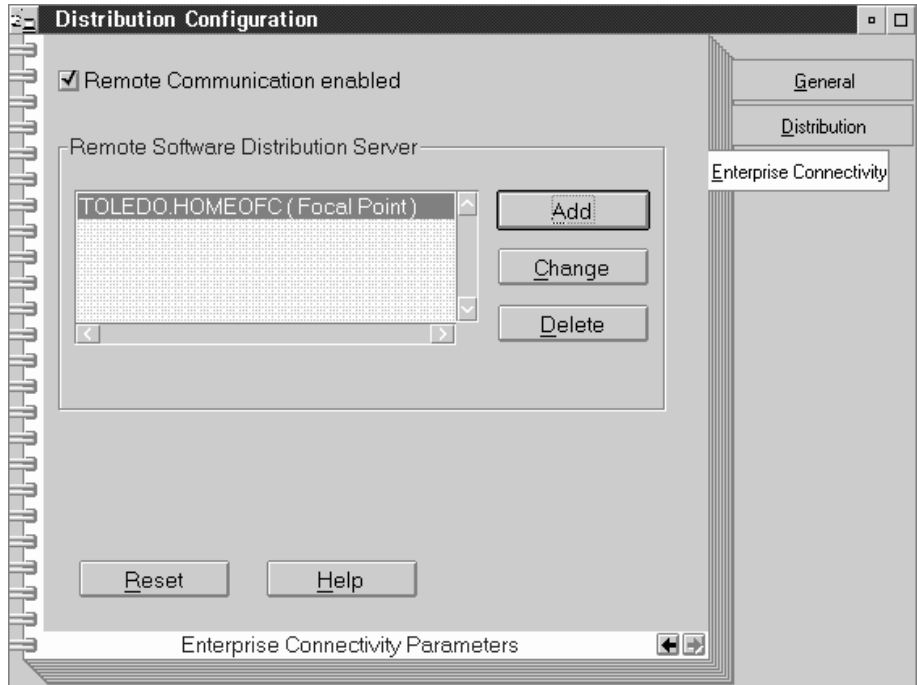


Figure 17. Enterprise connectivity page with remote server added

Close the notebook to save the configuration parameters.

- 10 Installation continues with the window shown in Figure 18.

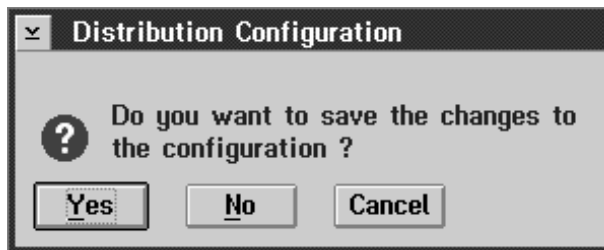


Figure 18. Configuration confirmation window

Select **Yes** to activate the configuration and continue installation.

- 11 Installation ends. A pop-up message is displayed telling you that the new configuration will be used when the program is restarted. Select **OK**.

- 12** Select **Exit** in the bottom right corner of the screen. Shut down and restart the system. The TME 10 Software Distribution icon is added to the desktop.

Logging on as the Root User

During installation, TME 10 Software Distribution defines the user *root* and configures it as the only user on both the server and any client. *Root* can be used to access the system initially and perform configuration tasks.

When the system defines the user *root*, it does not assign any password. *Root* can log on to the TME 10 Software Distribution server using the user ID only. For security purposes, at the TME 10 Software Distribution server, assign a password to *root*, which TME 10 Software Distribution will request at logon.

To assign a password to *root*, at the C: command prompt, enter the command:

```
nvdm updpwd root - n <pwd/pwd>
```

where <pwd> is the password (you must type it twice). It must be between six and eight characters long.

Assigning a Password to the User Root

If you assign a password to the user root, you must add this password to the following icons and files:

- The Start and Stop TME 10 Software Distribution Server icons, which you select from the TME 10 Software Distribution folder.
- The Start TME 10 Software Distribution Server icon, which you select from the OS/2 System - Icon View folder.
- The FNDUPD.CMD file
- The FNDINV.CMD file
- The FNDMIG.CMD file
- The FNDRCH.CMD file

If you do not perform these updates the product does not start correctly at the automatic startup.

To add the root's password to the various Start and Stop TME 10 Software Distribution icons, perform the following steps:

1. Locate the icons:

- In the TME 10 Software Distribution folder you will find a Start TME 10 Software Distribution Server icon and a Stop TME 10 Software Distribution Server icon
- To find the Start TME 10 Software Distribution Server icon in the OS/2 System - Icon View folder you should:
 - a. Click on the OS/2 System icon. The OS/2 System icon View folder appears

b. Click on the Startup icon. The Startup - Icon View folder appears

2. For each icon in turn, click on the icon with the right mouse button, and select Properties. The icon Properties window appears.

3. Select the Program tab.

4. Update the **Optional Parameters** field by adding the -p parameter as follows:

```
start -u root -p <password>
```

where <password> is the root's password.

To update the FNDUPD.CMD, the FNDINV.CMD, the FNDMIG.CMD, and the FNDRCH.CMD, files stored in the SOFTDIST\BIN directory, edit the files and enter the -p <password> parameter in the line that contains the nvdm command.

Using the APPC Configuration SmartGuides to Connect Over SNA

The APPC Configuration SmartGuides included with TME 10 Software Distribution provide an easy way to configure an APPC connection between the OS/2 server and:

- NetView DM for MVS
- TME 10 Software Distribution for AIX
- Other TME 10 Software Distribution OS/2 servers and clients

To configure an APPC link between a software distribution server and a partner, you configure a connection between two transaction programs that reside on the two systems.

The transaction programs are supplied by TME 10 Software Distribution for the OS/2 servers and clients, and by NetView DM for MVS and TME 10 Software Distribution for AIX for those platforms.

Every time the APPC connection is established, a remote conversation can take place between the two transaction programs; the programs take turns sending requests and waiting for their replies.

When you configure an APPC connection, you define two logical units of type 6.2, one per system, that provide the path through which the conversation can take place. You define to each system the name of the logical unit and transaction program for the local system and for the remote system (partner). You also define the mode in which the conversation takes place.

The SmartGuides help you to create the APPC configuration for your server. You enter the required values, and the tools take care of using the values where they are needed.

For APPC communication with another OS/2 server, an OS/2 client, or TME 10 Software Distribution for AIX, the SmartGuides generate Personal Communications APPC configurations. The *APPC With NetView DM for MVS* SmartGuide goes a step further. It can also create host definition files, which you can

send to your host administrator, and it does a minimal configuration for TME 10 Software Distribution.

You can find the SmartGuides in the **APPC Configuration SmartGuides** folder located in the main product folder. There are two icons, as shown in Figure 19:

- APPC with NetView DM for MVS
- APPC with Other Servers and Clients

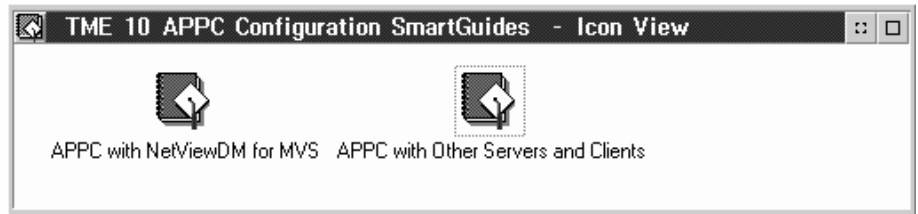


Figure 19. SmartGuides - icon view window

These icons start the corresponding SmartGuides.

In each SmartGuide, you first supply a configuration name. The tool uses it to name the configuration files it creates. Next, you provide the requested names and addresses, first for the server, then for the partner(s) with which the APPC connection will be established.

After you have finished entering the configuration information, the SmartGuide creates a response file to generate the Personal Communications configuration and optionally install Personal Communications; this response file can later be installed by the SmartGuide with the Personal Communications CMSETUP command.

The *APPC With NetView DM for MVS* SmartGuide can also generate:

- A list file with VTAM definitions and one with NCP definitions; you can send them to your host administrator to generate a definition for the TME 10 Software Distribution server.
- A list file with the NetView DM for MVS node definition for the server; you can send it to your host administrator to generate a definition for the TME 10 Software Distribution server.
- An update for the TME 10 Software Distribution configuration notebook defaults file, so that the Mode and Local LU Alias names will appear as defaults in the Remote Server Connection page of the notebook.

After the files have been created, you can choose to review them and change some of the preset keyword values.

Finally, you can have the SmartGuide install the response file automatically to generate the Personal Communications APPC configuration files. Alternatively, you can just exit the tool and have it installed at another time by reentering the configuration name.

Configuring for Remote Server Communication

In the *APPC with NetView DM for MVS SmartGuide*, if you generate the response file but exit the tool without installing it, you will have to run CMSETUP /R by hand. The response file, named *ConfigurationName.RSP* is saved in the SOFTDIST\BIN subdirectory; open an OS/2 session and from that subdirectory, enter:

```
CMSETUP /R ConfigurationName.RSP
```

In the *APPC With NetView DM for MVS SmartGuide*, you can also install the update to the TME 10 Software Distribution configuration.



1. The SmartGuides generate only new Personal Communications APPC configurations; they cannot update existing configurations.
2. When you install the Personal Communications response file, the resulting configuration files are added to the CMLIB directory. You can make the new configuration the current one through CMSETUP. Any existing configurations you may have are not deleted, but you may have to add additional features manually.

Configuring TCP/IP for Communication with a Remote Server

This section describes how to complete TCP/IP configuration on your TME 10 Software Distribution server to allow communication with a remote server. You need to perform these tasks only once, when TCP/IP is installed.

You must provide the following information:

- The name of the network interface you are using, for example, token ring or Ethernet.
- The host name for the workstation that you are configuring. This is the TCP/IP name by which this node is known.
- The Internet address of the node.
- The Internet address mask to define subnetworks if appropriate.

If you are in doubt about values for these fields, use the TCP/IP configuration online help.



In many cases, TME 10 Software Distribution is installed on a working system. If the workstation is already using TCP/IP, you do not need to perform these configuration steps.

Configuring at the Remote Server Site

To enable communication between an OS/2 distribution server and a remote server, follow these steps at the remote server site:

- 1 Edit the STS routing table to identify the TCP/IP connection between the remote server and your distribution server. Figure 20 shows an example of the STS routing table for a remote server (in this case AIX).

```
1
# This table provides STS routing information for
# TCP/IP routes.
# This file should be stored as /usr/lpp/netviewdm/db/routetab

NETWORK PROTOCOL:  TCP/IP

#
# Destination      Connection      Hop
# Address                               Count
#
OS2SRV1.OS2SV1      OS2SRVR          10
```

Figure 20. Example of an STS Routing Table for TCP/IP routes on an AIX remote server

Be sure NETWORK PROTOCOL is TCP/IP.

The format of the destination address in the STS routing table is *domain address.target address*. To get the domain address and target address, on the TME 10 Software Distribution distribution server, enter the command:

```
nvdm lstg -l
```

In this example, domain address is OS2SRV1 and target address is OS2SV1.

The connection name is any name you choose (in this example, OS2SRVR).

- 2 Create an STS connection configuration file. The name of the file must be the connection name you specified in the STS routing table (in this example, OS2SRVR). Figure 21 on page 70 is an example of an STS configuration file for a remote server.

Updating Your Installation

```
# STS CONNECTION CONFIGURATION FILE FOR CONNECTION OS2SRVR
#
# This connection is used to handle transmissions between
# TME 10 Software Distribution for AIX
# and TME 10 Software Distribution for OS/2
# using STS across TCP/IP.
#
# This file should be stored as /usr/lpp/netviewdm/db/routetab

PROTOCOL:                TCP/IP
TYPE:                     STS
REMOTE SERVER NAME:       OS2SRVR
```

Figure 21. Example of an STS Connection Configuration File for a remote server

Set REMOTE SERVER NAME to the workstation name of the remote TME 10 Software Distribution distribution server.

- 3 Now add the remote TME 10 Software Distribution distribution server as a target. Enter the command:

```
nvdm addtg workstation name -b server-y os/2. ►
-s target address -tp tcp:hostname
```

- 4 Enter the command:

```
nvdm rld
```

Updating Your Installation

You may need to update your installation, for example to add a component you had not installed previously, or you may wish to uninstall TME 10 Software Distribution. To do so, click on the **Installation Utility** icon in the product folder. The Installation and Maintenance window is displayed. Use the **Action** pull-down menu, as shown in Figure 22 on page 71, to specify what you want to do.

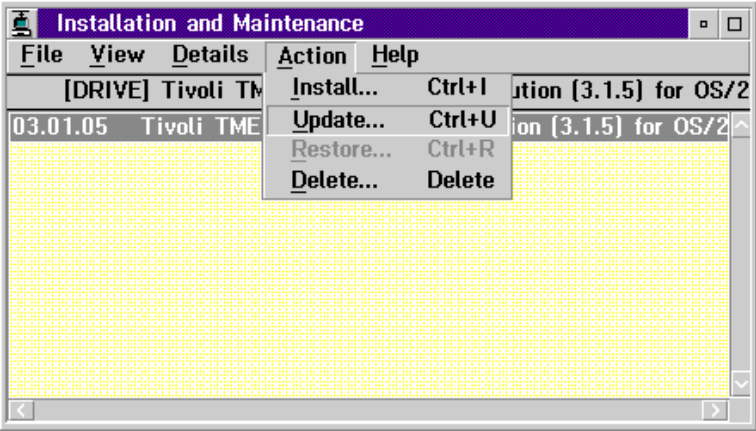


Figure 22. Installation and maintenance window

Use **Install...** to return to the interactive installation process and install components that were previously not installed.

Use **Update...** to install new versions of components and fixes.

Use **Delete...** to uninstall the product.

Be sure to stop the distribution server before making any changes to your installation.

Updating Your Configuration

To modify your configuration, click on the **Configuration** icon in the product folder to reopen the configuration notebook.

Chapter 10. Editing the Base Configuration File

The first configuration activities you perform on a newly installed TME 10 Software Distribution, Version 3.1.5 for OS/2 system are at the distribution server. First, you must define it as a target. You then work from it to perform the configuration activities required to set up your network.

Basic distribution server configuration is performed during the installation process and the information is stored in the base configuration file. This section describes the format of this file and the values that you can enter in its fields.

The base configuration file, `c:\SOFTDIST\nvdm.cfg`, contains the system parameters for controlling TME 10 Software Distribution, Version 3.1.5 for OS/2. The `nvdm.cfg` file generated at installation does not contain all possible parameters; rather, it contains a default subset of them. You can add or remove parameters and change their values by editing the file using a text editor.

The file is stored in a fixed text format. Each line starts with one of the keywords described in Table 12. Keywords must be entered in uppercase and followed by a colon. Each keyword, except for `PROTOCOL`, can be used only once. The order of keywords in the file is not important, and blank and comment lines can be included. Comment lines begin with a number sign character (`#`).

To make modifications to the file operative, you must stop and then restart the server using the icons in the product folder or the commands:

```
nvdm stop  
nvdm start
```

Note that if incorrect entries are entered in the base configuration file, the product may not start.

Table 12 (Page 1 of 9). Base Configuration File Parameters

Keyword	Description
API TRACE FILE SIZE	<p>The maximum size of the TME 10 Software Distribution API trace files, in bytes. When the trace file is full, it is automatically backed up and a new trace file is started.</p> <p>You can change this value but it should not normally be necessary. A large value requires more disk space. A small value degrades performance very slightly, because the trace file is backed up more often.</p>

Table 12 (Page 2 of 9). Base Configuration File Parameters

Keyword	Description
AUTHORIZE	<p>This keyword authorizes pull mode clients to install a software object or to execute a data file.</p> <p>NONE No target is authorized to install a software object or to execute a data file unless it is authorized through the <code>auth</code> command.</p> <p>ALL Any target is authorized to install a software object or to execute a data file unless it is unauthorized through the <code>unauth</code> command.</p> <p>The default is NONE.</p> <p>Set the AUTHORIZE keyword as soon as you have installed TME 10 Software Distribution, Version 3.1.5 for OS/2.</p>
AUTOMATIC TARGET INVENTORY	<p>This keyword causes clients, when they are added via command line or via GUI, to automatically perform a hardware and software inventory and store the results in the server database.</p> <p>Valid values are:</p> <p>YES The target inventory is performed when the target is added via command line or via GUI</p> <p>NO The target inventory is performed only when an automatic registration occurs. See the AUTOMATIC TARGET REGISTRATION keyword.</p> <p>The default is NO.</p>
AUTOMATIC TARGET REGISTRATION	<p>This keyword enables clients to automatically register themselves in the server database as one of the server's local targets. The registration is performed the first time a client connects to the server, if the client is not already registered.</p> <p>Valid values are:</p> <p>YES Enables targets to be registered at the server.</p> <p>NO Does not enable targets to be registered at the server.</p> <p>The default is NO.</p> <p>For automatic registration to be performed, the TARGET ADDRESS and TARGET MODE keywords must be set in the client's base configuration file, and the CONFIGURATION keyword in the client's base configuration file must be CLIENT.</p> <p>When an automatic target registration is performed an inventory is automatically scheduled.</p>

Table 12 (Page 3 of 9). Base Configuration File Parameters

Keyword	Description
BACKUP AREA	The name of the directory where software objects are stored when they are installed as removable. The default is C:\SOFTDIST\BACKUP. If you change the default path, you must copy all files into the new path; otherwise change control history information is lost. If the default path for the backup area is changed, it is not removed automatically if the product is uninstalled.
CONFIGURATION	<p>The configuration of your TME 10 Software Distribution, Version 3.1.5 for OS/2 node. The name is set up during installation and cannot be changed. It can be one of the following:</p> <p>SERVER_NO_COMMS Server with no remote connections to other servers</p> <p>SERVER_WITH_COMMS Server with remote connections to other servers</p> <p>The value of the field in the distribution server base configuration file can be displayed using the lsbs command.</p>
CONNECTION WINDOW DURATION	<p>The value you specify on this keyword becomes the default value for the d option on the connect command.</p> <p>It represents the amount of time, in minutes, that the connection window from the server to the client is to remain open.</p> <p>The default value is 60 minutes.</p>
DACA IDLE TIME	<p>The time in seconds after which an idle client-server connection is considered failed.</p> <p>Enter a value from 0 to 32 767. The default is 300.</p>
DACA RETRY TIME	<p>The time in seconds before a failed client-server connection is retried.</p> <p>Enter a value from 0 to 32 767. The default is 600.</p>

Table 12 (Page 4 of 9). Base Configuration File Parameters

Keyword	Description
FILE SYSTEM	<p>The name of the file system that the operating system supports. Based on this keyword, TME 10 Software Distribution chooses the local name when a request provides multiple local names.</p> <p>The default is one of the following numeric values, depending on the operating system:</p> <ul style="list-style-type: none"> 1 AIX 2 OS/2, Windows 3.11, DOS 3 NetWare 5 Windows 2000, Windows NT, Windows 98, Windows 95
INVENTORY PROGRAM	<p>The name of the program invoked when the inv command is issued. It creates the <code>fnswinv</code> file.</p>
LAN AUTHORIZATION	<p>Valid entries are:</p> <ul style="list-style-type: none"> 0 LAN address authorization is not required. 1 TME 10 Software Distribution, Version 3.1.5 for OS/2 validates LAN addresses on all LAN messages received by the distribution server. <p>If you specify LAN address authorization, you must supply the LAN address of each local target.</p> <p>The LAN authorization level can be set for individual work sessions using the updbs command at the command line interface or using the graphical interface. However, if you use one of the interfaces to change the value, it is not changed in the base configuration file, but is only modified for your current work session.</p>
LOG FILE SIZE	<p>The maximum size of the TME 10 Software Distribution, Version 3.1.5 for OS/2 message log file, in bytes. When the log file is full it is automatically backed up, and a new log file is started.</p> <p>You can change this value but it should not normally be necessary. A large value requires more disk space. A small value degrades performance slightly, because the log is backed up more often.</p>
MACHINE TYPE	<p>The operating system in use on your TME 10 Software Distribution, Version 3.1.5 for OS/2 node. The name is set up during installation and cannot be changed.</p>
MAX CONNECTIONS	<p>The maximum number of local targets allowed to process requests simultaneously.</p> <p>You can specify up to 50 connections. The default is 50.</p>

Table 12 (Page 5 of 9). Base Configuration File Parameters

Keyword	Description
MAX LOCAL TARGETS	The total number of local targets that can be configured for a server. The maximum number is 2000. The default is 512.
MAX REQUESTS	The maximum number of requests that can be active simultaneously. Enter a value from 1 to 65536. The default is 8192.
MAX SERVER CONNS	The maximum number of connections to remote server DACAs that can be open simultaneously. These STS connections receive requests from remote servers to perform operations. The MAX STS keyword, on the other hand, defines those STS connections that initiate operations to remote servers. Enter a value from 0 to 256. The default is 10.
MAX SERVER TARGETS	The total number of non-adjacent server targets that can be configured for a server. The total includes remote servers and single node targets. The MAX SNADS ROUTES and MAX STS ROUTES keywords are used to specify the number of these targets connected across either SNA/DS or STS connections. The maximum number is 2048. The default is 10.
MAX SNADS ROUTES	The maximum number of SNA/DS connections that can be defined. The maximum number is 800. The default is 20.
MAX STS	The maximum number of STS processes that can be simultaneously handling connections to remote distribution servers. These STS connections send requests for remote servers to perform operations. The MAX SERVER CONNS keyword, on the other hand, defines those STS connections that receive requests from remote servers. Enter a value from 0 to 256. The default is 10.
MAX STS ROUTES	The maximum number of STS connections that can be defined. The maximum number is 1600 minus the number of SNA/DS routes that have been defined (MAX SNA/DS ROUTES). The default is 30.

Table 12 (Page 6 of 9). Base Configuration File Parameters

Keyword	Description
MAX TARGETS	<p>The total number of local and/or remote targets that can be configured. The maximum number of targets is 40000.</p> <p>This value must be greater than the sum of MAX LOCAL TARGETS and MAX SERVER TARGETS.</p> <p>The formula MAX TARGETS minus MAX LOCAL TARGETS minus MAX SERVER targets determines the number of remote targets that can be configured at a server, for which there is no keyword in the base configuration file.</p> <p>The default for MAX TARGETS is 600.</p>
MAX USER INTERFACES	<p>The maximum number of simultaneous GUI and command line sessions. The default is 20.</p>
MESSAGE LOG LEVEL	<p>This field defines the log level that should be used by distribution clients before they establish a connection to the distribution server and discover the level configured for them there.</p> <p>Three log levels are available:</p> <p>M Minimal N Normal D Diagnostic</p>
PROTOCOL	<p>A transmission protocol used to communicate with other servers and clients. Enter a value using the following syntax:</p> <p><protocol name> <address></p> <p>Where:</p> <p><protocol name></p> <p>Is either TCP (for TCP/IP), IPX, NBI (for NetBIOS), or SNA.</p>
Continued overleaf	

Table 12 (Page 7 of 9). Base Configuration File Parameters

Keyword	Description
PROTOCOL (Continued)	<p><address></p> <p>If protocol name is TCP, specify:</p> <p><i>TCP hostname port_number max_connections</i></p> <p>If protocol name is NBI, specify:</p> <p><i>NBI NetBIOS_name adapter_number ► max_connections</i></p> <p>If protocol name is SNA, specify:</p> <p><i>SNA netid.luname.lumode</i></p> <p>If protocol name is IPX, specify:</p> <p><i>IPX IPX_address</i></p> <p>For a server, enter the IPX_address using the following syntax:</p> <p><internal_network> <server_address> ► <socket></p> <p>Where:</p> <p><Internal_network> is the value reported on the IPX Internal Network Number line of the output that you receive when you enter the config command on a NetWare console. It is 8 characters long.</p> <p><Server_address> is 000000000001 in a NetWare server. It is 12 characters long.</p> <p><socket> is 869F for a NetView DM for MVS server. The resulting string must be 24 characters long.</p> <p>For a client, enter the IPX_address using the following syntax:</p> <p><network_ID> <Mac_address> <socket></p> <p>Where:</p> <p><network_ID> is the value reported on the Lan protocol: IPX network... line of the output that you receive when you enter the config command on a NetWare console. On an OS/2 workstation it is the value that you find in the Lantran.log file. On a Windows NT workstation it is the value that you receive when you enter the Net config command. It is 12 characters long.</p> <p><Mac_address> is the physical address of the card you are using. It is 12 characters long.</p> <p><socket> is 869F for a NetView DM for MVS server. The resulting string must be 24 characters long.</p>

Editing the Base Configuration File

Table 12 (Page 8 of 9). Base Configuration File Parameters

Keyword	Description
REPOSITORY	The name of the repository where TME 10 Software Distribution, Version 3.1.5 for OS/2 stores its objects. The default repository is c:\SOFTDIST\repos. If the default repository path is changed, the repository is not removed automatically if the product is uninstalled.
SERVER	<p>The system name of this distribution server. It is set up automatically on the TME 10 Software Distribution server during installation.</p> <p>The server name is the text string that identifies the workstation name of the server machine. If you do not know this name, you can find it by looking at the TME 10 Software Distribution server base configuration file.</p> <p>If you want to attach a target to other distribution servers, so that you are able to administer and configure them, you must name those distribution servers here. This keyword can be used up to five times. The first instance defines the distribution server that provides the distribution server function for the distribution client being defined in this table, the other instances define distribution servers that can be administered from this distribution client.</p>
SERVICE AREA	The name of the directory for storing files and information pertaining to software objects that require activation. The directory should belong to a local file system mounted during startup. The default is c:\SOFTDIST\service. If you change the default path, you must copy all files into the new path; otherwise change control history information is lost. If the default path for the service area is changed, it is not removed automatically if the product is uninstalled.
STS IDLE TIME	<p>The time in seconds after which an idle STS connection is considered failed.</p> <p>Enter a value from 0 to 32 767. The default is 60.</p>
STS RETRY TIME	<p>The time in seconds before a failed STS connection is retried.</p> <p>Enter a value from 0 to 32 767. The default is 300.</p>
TARGET PASSWORD AUTHENTICATION	<p>This keyword activates target password authentication.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES • NO <p>The default is NO.</p>

Table 12 (Page 9 of 9). Base Configuration File Parameters

Keyword	Description
TRACE FILE SIZE	<p>The maximum size of the TME 10 Software Distribution, Version 3.1.5 for OS/2 internal trace files, in bytes. Two trace files are used and when one is full, tracing automatically switches to the other.</p> <p>You can change this value but it should not normally be necessary. A large value requires more disk space. A small value degrades performance very slightly, because the log is backed up more often.</p> <p>Refer to the online Message Reference file for information about the TME 10 Software Distribution, Version 3.1.5 for OS/2 internal tracing.</p>
WORK AREA	<p>The name of the directory where temporary work files are created and used. The default is c:\SOFTDIST\work. If the default path for the work area is changed, it is not removed automatically if the product is uninstalled.</p>
WORKSTATION NAME	<p>The name of the workstation that is running your TME 10 Software Distribution, Version 3.1.5 for OS/2 node. The name is set up during installation and cannot be changed.</p>

Figure 23 on page 82 is an example of the server base configuration file.

Editing the Base Configuration File

```
# BASE CONFIGURATION FILE
#
# This file should be stored as c:\SOFTDIST\nvdm.cfg
WORKSTATION NAME:      server1
SERVER:                server1
PROTOCOL:              TCP srv_hostname 729 50
REPOSITORY:            C:\SOFTDIST\REPOS
WORK AREA:             C:\SOFTDIST\WORK
BACKUP AREA:           C:\SOFTDIST\BACKUP
SERVICE AREA:         C:\SOFTDIST\SERVICE
CONFIGURATION:         SERVER_WITH_COMMS
MESSAGE LOG LEVEL:     D
LOG FILE SIZE:         2000000
API TRACE FILE SIZE:   2000000
TRACE FILE SIZE:       2000000
MACHINE TYPE:          OS/2
MAX USER INTERFACES:   20
MAX ATTEMPTS:           5
MAX TARGETS:           600
TARGET PASSWORD AUTHENTICATION: NO
AUTHORIZE:             NONE
LAN AUTHORIZATION:     0
AUTOMATIC TARGET REGISTRATION: YES
DACA RETRY TIME:       300
```

Figure 23. Example of a distribution server Base Configuration File

Chapter 11. Defining Users and Targets

This chapter provides the information needed to define users and targets in a network. You can find a complete description of how to define them using the command line interface in the online Command Reference for the product.

Defining Users

Users are defined at a database at the TME 10 Software Distribution server. A user definition includes the user's name and password, a description of the user, and security mechanisms that you can use to prevent unauthorized use of product functions and safeguard the privacy of sensitive data. They are:

- The definition of user profiles that enable the user to perform TME 10 Software Distribution, Version 3.1.5 for OS/2 functions. The system administrator can define profiles to be used by groups of users, or individual profiles for users. Three default authorization profiles are installed with the product; FNDADMN for system administrators, FNDBLD for change file builders, and FNDUSER for users.
- The definition of target access keys (TAK) that enable a user to work with specific targets.
- The definition of data access keys (DAK) that enable a user to work with specific objects, or data.
- The definition of the targets the user is authorized to login at

Defining Targets

In TME 10 Software Distribution, Version 3.1.5 for OS/2, the term *target* refers to all workstations in a network to which change control and distribution activities are directed. The following table contains a description of the parameters you use to define targets in a network.

Table 13 (Page 1 of 7). Target Definition Parameters

Parameter	Description
Name	Each target in a network has a unique name. When you communicate with a target, you use that unique name to identify it. A target name can contain up to 63 characters, excluding *, ?, and \.
Description	Text that describes the target. Up to 59 characters can be included in a description.

Table 13 (Page 2 of 7). Target Definition Parameters

Parameter	Description
Change management	<p>This parameter defines a target's <i>mode</i>. Targets can be configured in one of the following modes:</p> <p>Push mode Change control operations on a push mode target are controlled by the administrator at a TME 10 Software Distribution server or from a focal or manager target.</p> <p>Pull mode Change control operations on a pull mode target are controlled by the target's user.</p> <p>Manager A target configured in manager mode can perform change control operations on local targets and on remote targets. The remote administrator option must be installed on managers. Only targets whose type is server or single can be configured in manager mode. More than one target can be defined as a manager in a network.</p> <p>Focal A target configured in focal mode is a focal point for targets that define it as such. This means that the defining target routes all change control operation reports (both local and remote) to it. A focal target also acts as a manager. Only targets whose type is server or single be configured in focal mode.</p>

Table 13 (Page 3 of 7). Target Definition Parameters

Parameter	Description
Target address	<p>The address you assign to a target must be unique in its domain, and can be up to 8 alphanumeric characters long. For remote communication, the address of each target must be unique across the entire network. A complete address is composed of two parts, one of which is the target address you define here. The terminology used for the two parts of the address is different depending on the platform it runs on:</p> <ul style="list-style-type: none"> For TME 10 Software Distribution, Version 3.1.5 for OS/2 targets the terminology is the same as that used here. A complete target address is expressed as: <server name>.<target address> <p>If a target is a TME 10 Software Distribution server, both its <server name> and its <target address> must be the same.</p> <p>In SNA environments, these terms are also referred to as: <routing group name>.<routing element name> ► (RGN.REN)</p> <ul style="list-style-type: none"> For NetView DM for MVS and NetView DM/2 targets, the complete name is composed of: <Network ID>.<LU name> <p>Thus the <target address> entered in this field must correspond to a target's LU name.</p> <p>In most organizations, target names are allocated from a central authority (at the host processor) because this is the name to which change control instructions are sent. In a large network, it is often impossible to make this field meaningful. You may need to resort to the use of digits to produce unique names. If you are adding a remote target, its name must already be defined at the remote site.</p> <p>The address of a local target forms part of the global file name of files that are cataloged automatically for a target.</p>

Table 13 (Page 4 of 7). Target Definition Parameters

Parameter	Description
Target type	<p>A target must be one of the following <i>types</i>:</p> <p>Client</p> <p>Targets connected to a TME 10 Software Distribution server which have a TME 10 Software Distribution Client program installed.</p> <p>Clients can be either local or remote targets. Those in the same CC domain as their TME 10 Software Distribution server are referred to as <i>local</i> targets.</p> <p>Clients that belong to a different TME 10 Software Distribution domain are referred to as <i>remote</i> targets. Remote targets must be configured at all TME 10 Software Distribution servers they are to exchange files or change control requests with. Change control operations can be performed on remote targets only if the remote administration product option is installed on your system.</p> <p>Clients can be defined in push mode or pull mode.</p> <p>Clients can also be configured, or registered automatically, the first time they connect to a server. See “Automatic Client Registration” on page 89.</p> <p>Server</p> <p>Targets running TME 10 Software Distribution, Version 3.1.5 for OS/2 and which have the server option installed. TME 10 Software Distribution servers can perform change control and distribution operations on TME 10 Software Distribution clients in their software distribution domain. If they have the remote administrator product option installed, they can perform operations on remote targets.</p> <p>Server targets can be configured in push, pull, manager or focal mode.</p>

Table 13 (Page 5 of 7). Target Definition Parameters

Parameter	Description
Target type (cont.)	<p>Single</p> <p>Targets running TME 10 Software Distribution, Version 3.1.5 for OS/2 configured as a base system. Single-node targets can be used as preparation sites for software or as focal points to receive reports of change control operations. They can be configured in pull, push, manager or focal mode.</p> <p>Single-node targets can be accessed over TCP/IP and SNA/DS (APPC) networks. Change control operations can be performed on them if the remote administration option is installed on your system.</p> <p>Use this type also to define a NetView DM for MVS target.</p> <p>User Interface only (UI only) target</p> <p>Targets that can be used to run the TME 10 Software Distribution, Version 3.1.5 for OS/2 user interfaces. This type of target is useful when you have an environment in which more than one TME 10 Software Distribution server exists. It allows an administrator (a user belonging to the FNDADMN user group) to access all TME 10 Software Distribution servers from the same target either to perform administrative tasks or to schedule distributions to targets.</p> <p>A workstation configured in this manner is used to initiate change control on <i>other</i> targets, or to request distributions to and from the TME 10 Software Distribution server. User interface only targets cannot receive change control instructions from the same TME 10 Software Distribution server for which they are defined.</p> <p>You cannot define change control modes for UI only targets.</p>

Defining Targets

Table 13 (Page 6 of 7). Target Definition Parameters

Parameter	Description
Target OS	<p>The type of operating system installed on the target. It can be one of the following:</p> <p>AIX DOS HP_UX MAC NCR NETWARE OS/2 SCO SINIX SOLARIS SUNOS WINDOWS (for Windows 3.11) WIN95 (for Windows 95) WIN98 (for Windows 98) WINDOWS_NT WIN2K (for Windows 2000)</p>
Access key	<p>The key that allows users to access the target. A user can access a target only if he or she has been assigned the same access key as the one assigned to the target. Target access keys are defined by the system administrator.</p>
LAN address	<p>The target's burned-in LAN address. This parameter must be defined if the Validate LAN function is set to on. When it is on, if the target attempts to communicate with a TME 10 Software Distribution server, a check is performed to verify whether the LAN address stored in the target's database matches its actual address.</p> <p>The Validate LAN function can be set at either a TME 10 Software Distribution server or a target workstation. To change the setting from a target, the LAN address must already be defined in the database of the target.</p> <p>If you are adding a UI target only, a target's LAN address is automatically captured at the target itself, and stored at the TME 10 Software Distribution server. Leave this field blank, unless you need to change the automatically captured address.</p>
Server name	<p>For a client target, the name of the TME 10 Software Distribution server to which the target is linked.</p>

Table 13 (Page 7 of 7). Target Definition Parameters

Parameter	Description
Domain address	<p>This parameter is required for remote targets. The ID can be up to 8 characters long. The only valid characters are uppercase alphabets and numerics. The value you define depends on the type of target you are defining:</p> <p>TME 10 Software Distribution, Version 3.1.5 for OS/2 Server or Single Node Specify the target address specified for the target.</p> <p>Client Specify the server name of the TME 10 Software Distribution server to which the client is connected.</p> <p>Non TME 10 Software Distribution, Version 3.1.5 for OS/2 target If the target you are defining is <i>not</i> a TME 10 Software Distribution, Version 3.1.5 for OS/2 target, specify, for NetView DM/2 and NetView DM for MVS nodes, the Network ID of the remote network to which this target belongs. This value corresponds to the routing group name (RGN) part of the SNA/DS address of the target. Ask the administrator at the host processor what the value is.</p>
Password	The password used to access the target. It can have from 6 to 8 characters.

Automatic Client Registration

System administrators do not have to configure all client targets in a network individually at a server. They can be configured automatically, or autoregistered, the first time a client target connects to a server. For autoregistration to take place, the AUTOMATIC TARGET REGISTRATION keyword in a server's base configuration file must be set to YES (see Table 12 on page 73), and the TARGET ADDRESS and TARGET MODE keywords must be specified in the client base configuration file (see the *Installation and Configuration* manual for the TME 10 Software Distribution Clients).

When a client is automatically configured, its address and mode are inserted in the server database. Any other parameters for the client target must be specified manually using the graphical interface or the command line interface.

Automatic configuration is performed at all remote servers connected in a linear hierarchy to the first server that registers the target, as long as AUTOMATIC TARGET REGISTRATION is set to YES in each server's database. However, automatic target registration information is not routed from TME 10 Software Distribution, Version 3.1.5 for OS/2 servers to NetView DM/2 servers or NetView DM for MVS focal points.

Chapter 12. Configuring for Communication with MVS

Configuring Advanced Program to Program Communications with NetView DM for MVS implies configuring the communication subsystems required to support this type of link. This means that you must configure the following:

- Personal Communications/Communications Server APPC connection on the TME 10 Software Distribution server
- VTAM and NCP definitions at the MVS site
- NetView DM for MVS customization

The next sections explain how to configure these subsystems.

Remember that you can take advantage of the APPC Configuration SmartGuide, available online, to help you with some of these tasks. See “Using the APPC Configuration SmartGuides to Connect Over SNA” on page 66 for more information.

Configuring Personal Communications/Communications Server

You configure Personal Communications or Communications Server either through the respective user interfaces or through a response file to obtain a working APPC connection with NetView DM for MVS.

You can also use the APPC Configuration SmartGuide. The SmartGuide creates your Personal Communications/Communications Server configuration in an easy way. See “Using the APPC Configuration SmartGuides to Connect Over SNA” on page 66 for more information or try the APPC Configuration SmartGuide in the TME 10 Software Distribution for OS/2 Server window.

This section shows a sample Personal Communications node definition file for establishing a connection to NetView DM for MVS. The sample is organized and labeled by section for easier reading.

Defining the Local Control Point Profile

```
DEFINE_LOCAL_CP FQ_CP_NAME(ITIBM0PC.PPE0539 )
                DESCRIPTION(Physical Unit 539)
                CP_ALIAS(PPE0539 )
                NAU_ADDRESS(INDEPENDENT_LU)
                NODE_TYPE(EN)
                NODE_ID(X'05D00539')
                NW_FP_SUPPORT(NONE)
                HOST_FP_SUPPORT(YES)
                HOST_FP_LINK_NAME(HOST0001)
                MAX_COMP_LEVEL(NONE)
                MAX_COMP_TOKENS(0);
```

Defining the Logical Link Profile

```
DEFINE_LOGICAL_LINK LINK_NAME(HOST0001)
                    DESCRIPTION(Connection to Host)
                    FQ_ADJACENT_CP_NAME(ITIBM0PC.MVSESA31)
                    ADJACENT_NODE_TYPE(LEN)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'400015701088')
                    ETHERNET_FORMAT(NO)
                    CP_CP_SESSION_SUPPORT(NO)
                    SOLICIT_SSCP_SESSION(YES)
                    NODE_ID(X'05D00539')
                    ACTIVATE_AT_STARTUP(NO)
                    USE_PUNAME_AS_CPNAME(NO)
                    LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                    EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                    COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                    COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                    SECURITY(USE_ADAPTER_DEFINITION)
                    PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Defining the Local Logical Unit

```
DEFINE_LOCAL_LU LU_NAME(LT0539A0)
                LU_ALIAS(LT0539A0)
                NAU_ADDRESS(INDEPENDENT_LU);
```

Defining the Partner Logical Unit Profile

```

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(ITIBM0PC.CTNM15G)
                   PARTNER_LU_ALIAS(CTNDM15G)
                   PARTNER_LU_UNINTERPRETED_NAME(CTNDM15G)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(NO);2

```

Defining the Partner LU Location Profile

```

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(ITIBM0PC.CTNM15G)
                           WILDCARD_ENTRY(NO)
                           FQ_OWNING_CP_NAME(ITIBM0PC.MVSESA31)
                           LOCAL_NODE_NN_SERVER(NO);

```

Defining the Mode Profile

```

DEFINE_MODE  MODE_NAME(LU62      )
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(NO)
             MAX_RU_SIZE_UPPER_BOUND(4096)
             RECEIVE_PACING_WINDOW(4)
             MAX_NEGOTIABLE_SESSION_LIMIT(32767)
             PLU_MODE_SESSION_LIMIT(1)3
             MIN_CONWINNERS_SOURCE(0)
             PACING_TYPE(FIXED)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);

```

Defining the TME 10 Software Distribution TP for Inbound Transmissions

```

DEFINE_TP  SNA_SERVICE_TP_NAME(X'21',008)
           PIP_ALLOWED(NO)
           FILESPEC(I:\IBMSDS2\BIN\FNDTC.EXE)
           CONVERSATION_TYPE(EITHER)
           CONV_SECURITY_RQD(NO)
           SYNC_LEVEL(EITHER)
           TP_OPERATION(QUEUED_AM_STARTED)
           PROGRAM_TYPE(BACKGROUND)
           INCOMING_ALLOCATE_QUEUE_DEPTH(255)
           INCOMING_ALLOCATE_TIMEOUT(INFINITE)
           RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

```

² Note that Parallel Session Support must be set to NO for NetView DM for MVS sessions.

³ Note that for MVS sessions the Mode Session Limit must be set to 1.

Defining Targets for Communication with MVS

Defining the TME 10 Software Distribution TP for Outbound Transmissions

```
DEFINE_TP SNA_SERVICE_TP_NAME(X'21',007)
          PIP_ALLOWED(NO)
          FILESPEC(I:\IBMSDS2\BIN\FNDTC.EXE)
          CONVERSATION_TYPE(EITHER)
          CONV_SECURITY_RQD(NO)
          SYNC_LEVEL(EITHER)
          TP_OPERATION(QUEUED_AM_STARTED)
          PROGRAM_TYPE(BACKGROUND)
          INCOMING_ALLOCATE_QUEUE_DEPTH(255)
          INCOMING_ALLOCATE_TIMEOUT(INFINITE)
          RECEIVE_ALLOCATE_TIMEOUT(INFINITE);
```

Starting the Attach Manager

```
START_ATTACH_MANAGER;
```

Defining Targets for Communication with MVS

To define a NetView DM for MVS system as a focal point, enter the following command:

```
nvdm addtg <target name> -s <target address>
          -b server -m focal -n <domain address>
```

For example, enter:

```
nvdm addtg CTNDM154 -s CTNDM154 -n ITIBM0PC -m focal -b server
```

To define a TME 10 Software Distribution, Version 3.1.5 for OS/2 target at MVS, on the GIX **Define Node** panel enter the following information. Note that the <domain address> and the <target address> to be inserted respectively in the **Rgn** and **Ren** fields are the same value.

1 Node class . A0	Required
2 Status . . . 2	1 = Production 2 = Parallel 3 = Test
3 Logical unit LT0539A0	Required (Logical unit name)
4 Logon mode . LU62 ____	Logon mode name
5 Linetype . . 1	1 = Leased 2 = Switched
6 Rgn. <domain address>	Network identification
7 Ren. <target address>	CP Logical unit name
8 Notes . . . 2	Enter 1 if you want additional node info
9 Profile. . . 2	Enter 1 if you want to change node profile
10 Server name. <target address>	Server name
11 Timzoffs . . +00	Time Zone offset. Any value from -12 to 12

Figure 24. Defining a TME 10 Software Distribution, Version 3.1.5 for OS/2 node at MVS

Configuring at the NetView DM for MVS Site

This section describes the configuration procedures you must perform to enable communication between a TME 10 Software Distribution server and NetView DM for MVS.

Configuring NCP and VTAM

To define a TME 10 Software Distribution node to VTAM, you need to work with your NetView DM for MVS system administrator to agree on naming conventions. You need to:

- Define the TME 10 Software Distribution physical and logical units (see “Physical and Logical Unit Definitions”).
- Define the parameters in a logon mode table for the sessions between the NetView DM for MVS system and TME 10 Software Distribution (see “Logon Mode Table Definition” on page 103).

Examples are shown in the following section, but they do not provide an exhaustive definition of how to configure VTAM and NCP. Consult your *VTAM Resource Definition Reference* where necessary.

Physical and Logical Unit Definitions

Each TME 10 Software Distribution that is connected *directly* to NetView DM for MVS (that is, not through an intermediate node) requires an independent LU to be defined to VTAM and also to NCP if it is being used. A physical unit is also required for any intermediate node that provides a connection to a TME 10 Software Distribution node.

The following sections provide examples of configuring direct connections. They are for the following configurations:

- Connection to a 3745 attached to the LAN (“Connection through a 3745 Gateway”)
- Connection through a token-ring gateway (“Connection through a Token-Ring Gateway” on page 97)
- Direct connection using an SDLC link (“Direct Connection Using an SDLC Link” on page 100)

In addition, an example of a logon mode table is shown. You can use this table for any of the three sample configurations.

The APPC Configuration SmartGuide provides personalized configurations for these types of connections, in addition to a connection through a 3172 VTAM gateway attached by token-ring or Ethernet.

Connection through a 3745 Gateway

This section gives the NCP and VTAM definitions necessary for TME 10 Software Distribution to communicate with NetView DM for MVS through a 3745 gateway. See Figure 25 on page 96.

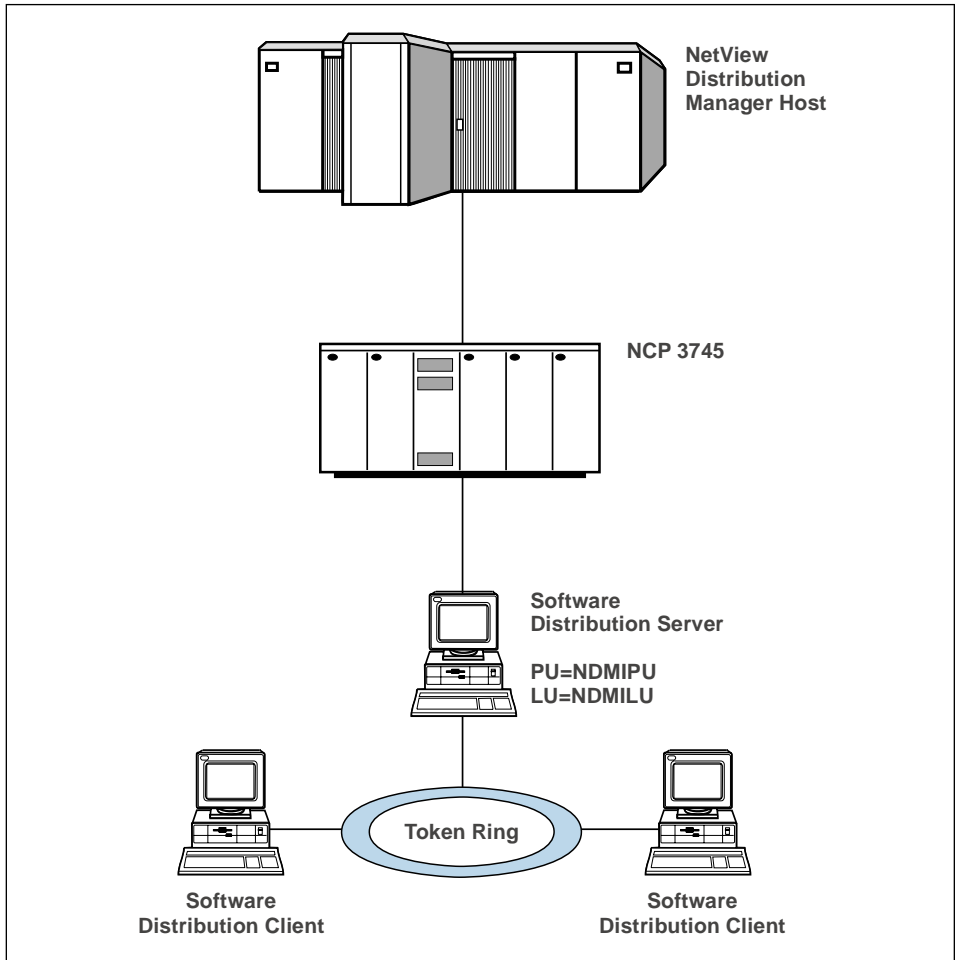


Figure 25. TME 10 Software Distribution and NetView DM for MVS using a direct link and a 3745

NCP Definition for Token-Ring Connection: The statements in Figure 26 show the NCP macros used to define the connectivity to the token ring and the PU/LU pairs required to support devices on a switched line. (TME 10 Software Distributions on a LAN token ring are supported as devices on a switched line.)

```

TO27TRPG GROUP ECLTYPE=PHYSICAL
TO27TRL1 LINE ADDRESS=(1092,FULL),LOCADD=400037271092,PORTADD=1
RCVBUFC=4095,MAXTSL=1108,TRSPD=16,ADAPTER=TIC2
TO27TRP1 PU ADDR=01,PUDR=NO,ANS=CONT
LUTIC2 LU LOCADDR=0,ISTATUS=INACTIVE
TO27TRG1 GROUP ECLTYPE=LOGICAL,AUTOGEN=24,PHYPORT=1,CALL=INOUT
    
```

Figure 26. NCP macros for Token-Ring connectivity

VTAM Switched Major Node Definitions: The statements in Figure 27 show the VTAM macros used to define the switched major node containing the PU and LU statements for the TME 10 Software Distribution node.

Note that LOCADDR=0 indicates an independent LU.

```
DIALNMD6 VBUILD TYPE=SWNET,MAXGRP=2,MAXNO=2

NDM6PU  PU ADDR=C1,IDBLK=071,IDNUM=00013,PUTYPE=2,ISTATUS=ACTIVE,
        DLOGMOD=NVDNMNORM,USSTAB=TPOUSS,MAXOUT=7,MAXDATA=265,    *
        PACING=0,VPACING=0,ANS=CONT,MODETAB=NDMLU62P

NDM6LU  LU LOCADDR=0,MODETAB=NDMLU62P,DLOGMOD=NVDNMNORM,
        ISTATUS=ACTIVE
```

Figure 27. VTAM macros for TME 10 Software Distribution connectivity

TME 10 Software Distribution Definitions: Table 14 shows the correspondence between VTAM and TME 10 Software Distribution parameters.

Table 14. VTAM and TME 10 Software Distribution Parameters, 3745 Connection	
VTAM Parameter	TME 10 Software Distribution Parameter
NETID=NETWK1	Network name
IDNUM=00013	Node ID
DLOGMOD=NVDNMNORM	Mode name
PU=NDM6PU	PU name
LU=NDM6LU	LU name for TME 10 Software Distribution
LOCADDR=0	Indicates that this is an independent LU
LOCADD=400037271092 (NCP parameter)	The address that is specified in the Personal Communications link definition

Connection through a Token-Ring Gateway

This section gives the 3174 and NCP/VTAM definitions necessary for TME 10 Software Distribution to communicate with NetView DM for MVS through a token-ring gateway. See Figure 28 on page 98.

Configuring at the NetView DM for MVS Site

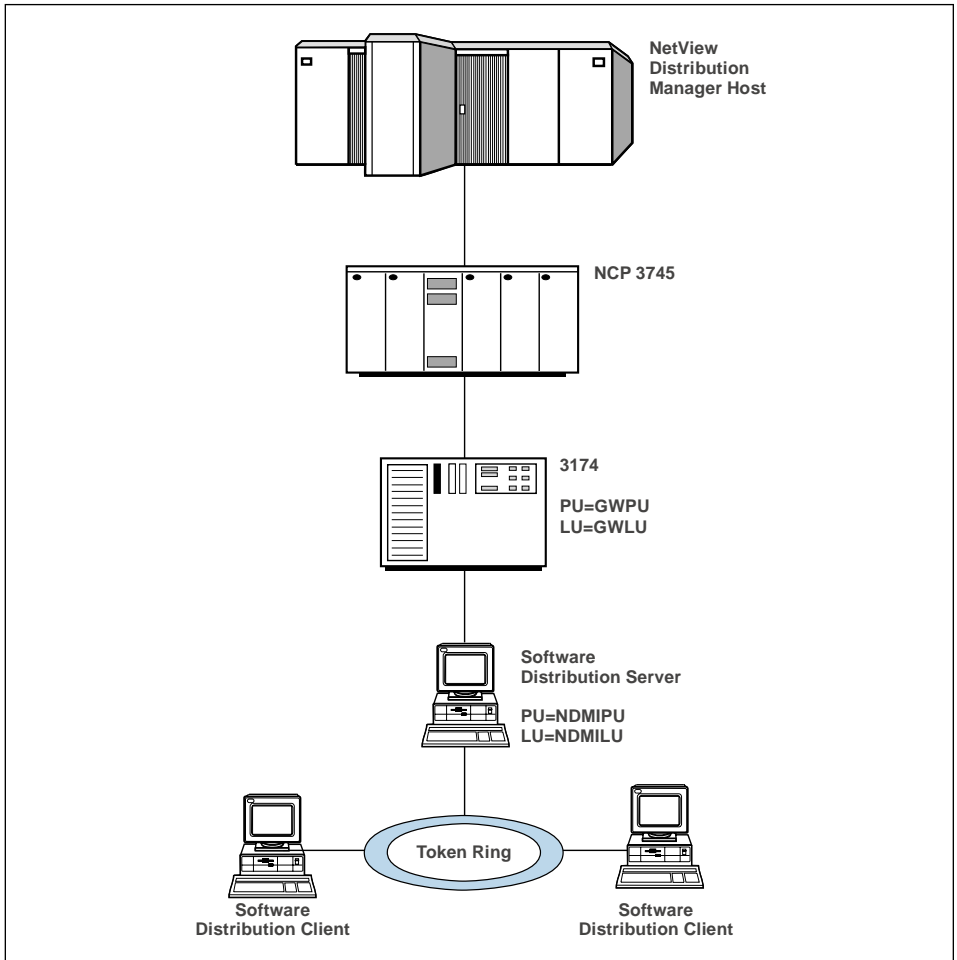


Figure 28. TME 10 Software Distribution and NetView DM for MVS using a Token-Ring gateway

The gateway can be any proprietary software or it can be a 3174. Instructions for configuring the gateway are not given here; refer to the documentation provided with the gateway.

NCP/VTAM Definitions: Figure 29 on page 99 shows sample NCP/VTAM definitions. They refer to the network shown in Figure 28.

GSDLC	GROUP	CLOCKNG=EXT, DIAL=NO, LNCTL=SDLC, REPLYTO=0.5, RETRIES=(19,4,5), TYPE=NCP
L002	LINE	ADDRESS=002, DUPLEX=FULL, ETRATIO=25, SPEED=9600, ISTATUS=ACTIVE, RETRIES=(7,2,2)
GWPU	PU	ADDR=C0, ISTATUS=ACTIVE, PACING=(1), PUDR=YES, PUTYPE=2, DISCNT=(NO)
GWLU	LU	LOCADDR=1, ISTATUS=ACTIVE, MODETAB=TPOMODE, DLOGMOD=SD82
NDM6PU	PU	ADDR=C1, MAXDATA=265, MAXOUT=7, PASSLIM=8, PUTYPE=2, SSCPFM=USSSCS, XID=YES
NDM6LU	LU	LOCADDR=0, ISTATUS=ACTIVE, MODETAB=NDMLU62P, DLOGMOD=NVDNMNORM, RESSCB=2, PACING=1

Figure 29. VTAM and NCP macros for connectivity through a gateway

Configuring at the NetView DM for MVS Site

The definition in Figure 29 contains the following:

- A PU for the gateway (GWPU)
- An LU for 3270 use at the gateway (GWLUI)
- A PU for the TME 10 Software Distribution node using the *same line* as that used to reach the gateway (NDM6PU)
- An independent LU for the TME 10 Software Distribution node (NDM6LU)

Note that this LU is defined under the PU for the TME 10 Software Distribution node. The LU has LOCADDR=0 to indicate an independent LU.

TME 10 Software Distribution Definitions: Table 15 shows the correspondence between VTAM and TME 10 Software Distribution parameters.

<i>Table 15. VTAM and TME 10 Software Distribution Parameters, Token-Ring Gateway Connection</i>	
VTAM Parameter	TME 10 Software Distribution Parameter
NETID=NETWK1	Network name
DLOGMOD=NVDNMORM	Mode name
PU=NDM6PU	PU name
LU=NDM6LU	LU name for TME 10 Software Distribution
ADDR=C1	Local station address

Direct Connection Using an SDLC Link

This section gives examples of the NCP and VTAM definitions necessary for a TME 10 Software Distribution node to communicate with NetView DM for MVS over a direct SDLC link. See Figure 30 on page 101.

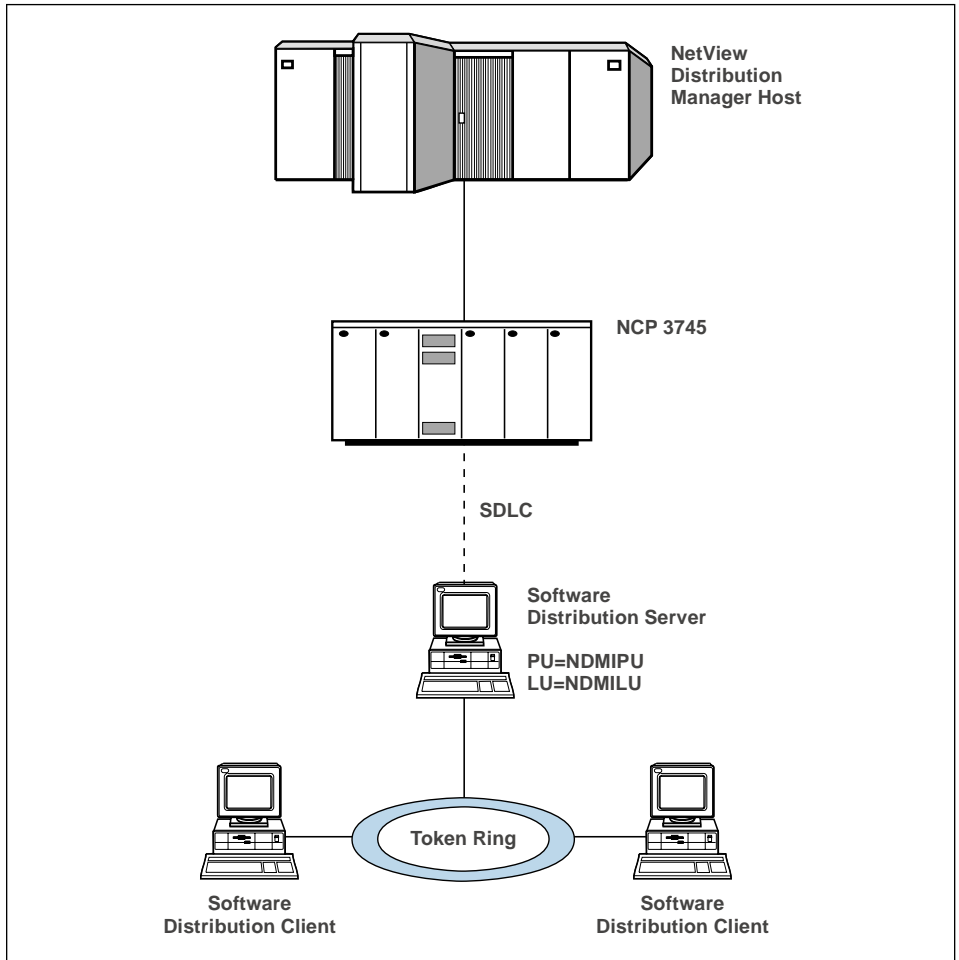


Figure 30. SDLC connection to NetView DM for MVS

NCP/VTAM Definition: Figure 31 on page 102 shows sample NCP/VTAM definitions. They refer to the network shown in Figure 30.

GSDLC	GROUP	CLOCKNG=EXT, DIAL=NO, LNCTL=SDLC, REPLYTO=0.5, RETRIES=(19,4,5), TYPE=NCP
L002	LINE	ADDRESS=002, DUPLEX=FULL, ETRATIO=25, SPEED=9600, ISTATUS=ACTIVE, RETRIES=(7,2,2)
NDM6PU	PU	ADDR=C1, MAXDATA=265, MAXOUT=7, PASSLIM=8, PUTYPE=2, SSCPFM=USSSCS, XID=YES
NDM6LU	LU	LOCADDR=0, ISTATUS=ACTIVE, MODETAB=NDMLU62P, DLOGMOD=NVDNMNORM, RESSCB=2, PACING=1

Figure 31. VTAM and NCP macros for connectivity using an SDLC link

The definition in Figure 31 contains the following:

1. A direct attachment on line L002 to the PU for the TME 10 Software Distribution node (NDM6PU)
2. An independent LU (LOCADDR=0) for access to the TME 10 Software Distribution node defined directly under the PU (NDM6LU).

TME 10 Software Distribution Definitions: Table 16 on page 103 shows the correspondence between VTAM and TME 10 Software Distribution.

Table 16. VTAM and TME 10 Software Distribution Parameters, SDLC Connection

VTAM Parameter	TME 10 Software Distribution Parameter
NETID=NETWK1	Network name
DLOGMOD=NVDNMNORM	Mode name
PU=NDM6PU	PU name
LU=NDM6LU	LU name for TME 10 Software Distribution
ADDR=C1	Local station address

Logon Mode Table Definition

The logon mode table is used to define the parameters for sessions between TME 10 Software Distribution and NetView DM for MVS. This table defines the logon mode NDMLU62P that is referred to in all the previous examples.

```

MODETAB

NDMLU62P MODEENT LOGMODE=NVDNMNORM,
          FMPROF=X'13',
          TSPROF=X'07',
          PRIPROT=X'B0',
          SECPR0T=X'B0:',
          COMPROT=X'50A1',
          ENCR=B'0000',
          RUSIZES=X'8585',
          PSNDPAC=X'03',
          SRCVPAC=X'03',
          SSNDPAC='00',
          PSERVIC=X'0602000000000000000000002400',
          TYPE=X'0',
          COS=COSNAME
MODEEND

```

Figure 32. Logon mode table

The parameters in Table 17 are specific for defining the LU session between TME 10 Software Distribution and NetView DM for MVS.

Table 17 (Page 1 of 5). Parameters for Defining the LU Session

Parameter	Description
LOGMODE	Specifies the entry name that is used to point to the set of session parameters in this logon mode table. You can use any entry name with up to eight characters; however, this must match the mode name specified within the configuration for TME 10 Software Distribution.

Table 17 (Page 2 of 5). Parameters for Defining the LU Session

Parameter	Description
FMPROF	Specifies the function management profile for this entry. Set FMPROF to X'13', which indicates that Function Management Profile 19 (X'13') rules are to be used for these LU 6.2 sessions. See <i>Systems Network Architecture Formats</i> .
TSPROF	Specifies the transmission services profile for this entry. Set TSPROF to X'07', which indicates that Transmission Services Profile 7 (X'07') rules are to be used for these LU 6.2 sessions. See <i>Systems Network Architecture Formats</i> .
PRIPROT	Specifies the primary LU protocols for this entry. Set PRIPROT to X'B0', which indicates the following protocols are to be used: <ul style="list-style-type: none"> • Multiple RU chaining • Immediate request mode • Definite or exception response.
SECPROT	Specifies the secondary LU protocols for this entry. Set SECPROT to X'B0'.
COMPROT	Specifies the common LU protocols for this entry. Set COMPROT to X'50A1', which indicates that the following protocols are to be used: <ul style="list-style-type: none"> • Segmenting • FMH allowed • Brackets used • CEB used • No alternate code set • BIND RSP not held • HDX-FF • Symmetric recovery • SLU=winner • HDX-FF reset is SEND for PLU and RCV for SLU
ENCR	Specifies the type of cryptography to be used with the VTAM data encryption facility. Set ENCR to B'B0000', that is, no cryptography because VTAM does not support encryption for LU 6.2 sessions.

Table 17 (Page 3 of 5). Parameters for Defining the LU Session

Parameter	Description
RUSIZES	<p>Specifies the maximum length of data (request units or RU) in bytes that can be sent by the primary LU and the secondary LU when they are in session with each other.</p> <p>RUSIZES is represented by four hexadecimal digits. The two leftmost digits apply to the secondary LU and the two rightmost digits apply to the primary LU. The format is the same for both sets of digits: in each set, the first digit is the mantissa (m) and the second digit is the exponent (n) in the formula $m \times 2^n$. The mantissa must be in the range X'8'–X'F'. The exponent must be in the range X'0'–X'F'.</p> <p>This formula is then used to calculate the maximum RU sizes that can be sent by the primary or secondary LU. For example, RUSIZES=X'858C' specifies that the secondary LU can send an RU of maximum length 8×2^5 (or 256) bytes and the primary LU can send an RU of maximum length 8×2^C (or 32768) bytes.</p> <p>TME 10 Software Distribution supports all RU sizes. However, you should note the following guidelines when setting up the logon mode table and when configuring Personal Communications:</p> <ul style="list-style-type: none"> • Small RUs make poor use of your data transfer mechanism. Transfer times, especially for large files, are considerably increased by using small RUs. The smallest RU size that you should consider is 256 bytes. • Large RUs can cause saturation at the receiver if the data cannot be processed as fast as it is received. • Some SNA connections, especially slow ones like public telecommunication lines, are better suited to small RUs (for example 256 or 512 bytes). • TME 10 Software Distribution supports any legal RU size between 256 and 3840 bytes. Any BIND proposing a size smaller than 256 is rejected by TME 10 Software Distribution; any BIND proposing a size greater than that configured is negotiated downward.

Table 17 (Page 4 of 5). Parameters for Defining the LU Session

Parameter	Description
	<p>Therefore, RUSIZES is best set to X'F8F8' (4 KB) maximum and X'8585' (256 bytes) minimum. For a TME 10 Software Distribution node connected directly through a token ring to a channel-attached NCP, an RU size of 3840 is expected to be optimal. For other attachments, link speeds and connectivity must be considered when you are selecting the optimal value.</p> <p>NetView DM for MVS does not check the RUSIZES value for NetView DM for MVS-to-TME 10 Software Distribution sessions. In the sample logmode definition, RUSIZES is set to X'8585' (256 bytes) for NetView DM for MVS sessions. These values should be consistent with those defined to the TME 10 Software Distribution node.</p>
PSNDPAC	<p>Specifies the primary send pacing count. This value is not checked by TME 10 Software Distribution. If PSNDPAC is omitted, PSNDPAC=X'00' is the default.</p>
SRCVPAC	<p>Specifies the secondary receive pacing count. If SRCVPAC is omitted, SRCVPAC=X'00' is the default.</p> <p>SRCVPAC must be appropriate to the RU size selected for the primary LU, and is determined by the formula:</p> $(2n - 1) \times \text{primary send RU size} \leq 4\text{KB}$ <p>where n is the SRCVPAC value.</p> <p>For example, SRCVPAC=X'01' if the two rightmost digits of RUSIZES are X'F7'. If the RUSIZES chosen is X'8585', SRCVPAC can be as high as X'08'.</p> <p>The value of SRCVPAC greatly influences throughput. Higher values of this parameter ensure better throughput. Set SRCVPAC as high as the calculation allows.</p> <p>Note that SRCVPAC can be negotiated downward but never upward.</p>
SSNDPAC	<p>Specifies the secondary send pacing count. TME 10 Software Distribution has no dependencies on this value. It is used as specified. SSNDPAC=X'00' is the default.</p>

Table 17 (Page 5 of 5). Parameters for Defining the LU Session

Parameter	Description
PSERVIC	Specifies the presentation services profile for this entry. Set PSERVIC to X'060200000000000000002400' which indicates that the following will be used: <ul style="list-style-type: none"> • LU 6.2 • No attach security • Sync level=confirm • PLU reinitiates • Parallel sessions are not supported.
TYPE	Specifies the type of BIND command for this entry. Set TYPE to X'0', which means that the secondary LU can support a negotiable BIND.
COS	Specifies the name of an entry in a class of service table to be used for sessions established with this logon mode. Because it is a <i>batch</i> mode, you should use an entry that specifies low-priority virtual routes so as not to interfere with interactive traffic.

Configuring NetView DM for MVS

NetView DM for MVS requires two specific customization steps to manage TME 10 Software Distribution nodes.

- 1 Define node types with change management entry point (CMEP) functional capabilities for TME 10 Software Distribution TME 10 Software Distribution servers.

You can use the APPC Configuration SmartGuide to obtain these definitions.

You do not have to define the TME 10 Software Distribution clients. They define themselves automatically when TME 10 Software Distribution is installed.

- 2 When NetView DM for MVS has been configured to manage this type of node, prepare the specific network definition for both directly and indirectly connected nodes.

If a node type with CMEP functional capabilities has not been defined, then you must perform a new run of the installation macros.

Prepare the following for the NetView DM for MVS stage 1 installation job:

- NDMNODE
- NDMTCP
- NDMCP
- A transmission profile

Prepare a macro defining the characteristics for a TME 10 Software Distribution server.

Configuring at the NetView DM for MVS Site

In NetView DM for MVS, specify a node type for TME 10 Software Distribution servers and declare it to have change management entry point (CMEP) functional capabilities.

In the example in Figure 33, the node type for TME 10 Software Distribution servers has been assigned the name **NDM6**.

Now prepare the following macros:

- 1 An NDMNODE macro to define a node type with CMEP functional capabilities for TME 10 Software Distribution nodes (see Figure 33).

```
NDMNODE TYPE=NDM6,  
        LOGM=NVDNMNORM,FUNC=CMEP,  
        XMFUNC=(SEND, RETR, DELE),  
        RESTYPE=(0060,0070,0080,0100,0120,0220,0230,0240,0250)
```

Figure 33. Sample NDMNODE macro for TME 10 Software Distribution Server

The variables defined are shown in Table 18.

Table 18 (Page 1 of 2). NDMNODE Variables	
Parameter	Description
LOGM	Set this value to the name of a logon mode table that defines the session parameters for communication between TME 10 Software Distribution and NetView DM for MVS. (Refer to the example in Figure 32 on page 103.)
FUNC	The functional capability must be defined as CMEP. A CMEP node can manage changes to itself and has limited ability to manage changes to other nodes.
XMFUNC	This can be set to (SEND, RETR, DELE) for a TME 10 Software Distribution node. This parameter defines the transmission-function authorization parameters. That is, it defines those functions that the TME 10 Software Distribution nodes are allowed to initiate against NetView DM for MVS. TME 10 Software Distribution nodes support sending, retrieving, and deleting files but cannot initiate change control commands.

Table 18 (Page 2 of 2). NDMNODE Variables

Parameter	Description
RESTYPE	<p>This parameter gives the data object classifications that the TME 10 Software Distribution node is authorized to work with. TME 10 Software Distribution can work with all of the defined resource types, so you should list them all unless you have some reason to limit the types of files that you want a particular node to have permission to transfer.</p> <p>This parameter must be present if the XMFUNC parameter is specified. Refer to the <i>NetView DM for MVS Installation and Customization Guide</i> for possible values.</p>
SFUNC	<p>This parameter lists the functions that may be issued to the TME 10 Software Distribution node. The default is the full set (SEND, RETR, DELE, REMO, ACTI, INIT, ACCE, INST, UNIN).</p> <p>This parameter need not be specified, because TME 10 Software Distribution supports all of the functions. To limit the functions that can be issued by NetView DM for MVS to a particular node, you must use this parameter and specify only those functions that are allowed.</p>

- 2** An NDMTCP macro specifically for the TME 10 Software Distribution application. A sample macro highlighting the lines that are significant for TME 10 Software Distribution is shown in Figure 34 on page 110.

```

NDMTCP APPLID=(RAKADT03,*),
      IAPPLID=(RAKADI03,*),
      IPLUNAM=NONE,      (LU NAME OF DEFAULT IOF PRINTER)
      OPCTL=NETV,        (SELECTED MESSAGES ARE ROUTED TO NV)
      ROUTCD1=2,
      ROUTCD2=2,
      DSCD1=6,
      DSCD2=6,
      RESWAIT=300,      (SECONDS NDM WAITS FOR LU62 RESPONSE)
      STALINE=1,        (DEFAULT IS NO)
      AUTOSTR=NO,        (NO IS THE DEFAULT)
      AUTOEND=NO,        (NO IS THE DEFAULT)
      RETRY=3,          (RETRY COUNT FOR INTERRUPTED SESSION)
      RETINT=30,         (TIME WAITED BY TCP BEFORE A RETRY)
      APPC=YES,          (YES IF NDM IS TO HAVE LU6.2 SESSION)
      MAXTASK=(4,1),    (CONCURRENT SESSION TOTAL,SWITCHED SESS)
      SWDLY=5,           (SECS. WAITED BEFORE VTAM SESSION RETRY)
      SWRTRY=3,          (NUMBER OF VTAM SESSION RETRIES)
      DDPREQ=YES,        (FORCED TO YES IF APPC=YES)
      RESYNCH=4,         (4098K BYTE BLOCKS BETWEEN CHECKPOINTS)
      MSGINFO=2,         (ALL MSGS GO TO SYSPRINT/IOF/CONSOLE)
      HOPCNT=5,          (NO. OF NODES AN LU6.2 MSG. CAN HOP)
      AUTEXIT=NDMEXIT,   (USER-EXIT TO POINT TO RACF FOR IOF)
      NDCCAPI=NO,        (DO NOT HAVE API FEATURE INSTALLED)
      QMSURPT=NO,        (DEFAULT. YES IF U HAVE USER APPLS)
      SUFFIX=18
    
```

Figure 34. Sample NDMTCP macro for TME 10 Software Distribution

The significant parameters are described in Table 19.

Table 19 (Page 1 of 2). NDMTCP Parameters	
Parameter	Description
RESWAIT	Defines the number of seconds NetView DM for MVS waits for a response from a TME 10 Software Distribution node.
APPC	This must be set to YES to indicate that APPC is in use.

Table 19 (Page 2 of 2). NDMTCP Parameters

Parameter	Description
HOPCNT	<p>A distribution is passed from node to node across the SNA network. Each time it is passed is called a <i>hop</i>.</p> <p>The hop count is the maximum number of hops that can be included in a single routing chain from NetView DM for MVS. The distribution must reach its destination within this number of hops.</p> <p>This parameter is provided to prevent messages looping forever in large networks with conflicting routing instructions at different nodes. In simple networks, assign this parameter a small number.</p>

- 3** An NDMCP macro to define connection profiles that group LU 6.2 logical units that have the same attributes and connection capabilities.

If your TME 10 Software Distribution workstation is connected by a line that allows the workstation to establish a session (for example, a leased line):

- a** Define a connection profile with the polling parameter set to NO.
- b** Assign the LU name of the TME 10 Software Distribution workstation to that connection profile.

In this way, TME 10 Software Distribution will not poll for a reply for a host-initiated request. An example is shown in Figure 35.

```
NDMCP  CPNAME=CP02,
        POLLING=NO
```

Figure 35. Example of an NDMCP macro

- 4** A transmission profile that groups nodes to connect them to the central site using the same type of line. Grouping nodes in this way optimizes line usage. For example, if a multipoint line is used to connect eight nodes, they could be grouped by a transmission profile that specifies that no more than three concurrent transmissions can take place against them, ensuring better load balancing across different lines.

Each transmission profile can have its own retry specifications.

An example of the NDMTP customization macro that is used to define a transmission profile is shown in Figure 36.

```
NDMTP  TPNAME=TPNDM2,TPTYPE=L,MINGR=27,MAXN=50
```

Figure 36. Example of an NDMTP macro

Configuring at the NetView DM for MVS Site

where:

- TPNAME is the name of the transmission profile
- TPTYPE specifies the connection type of the nodes (in this example, a leased line)
- MINGR specifies the number of transmission tasks that the TCP can grant to the transmission profile
- MAXN specifies the maximum number of transmission tasks that can be active at the same time

Chapter 13. Configuring Personal Communications/Communications Server

This chapter includes sample output files for Personal Communications 4.x configurations:

- For a SNA/DS connection between a TME 10 Software Distribution for OS/2 server and another TME 10 Software Distribution for OS/2 server, a TME 10 Software Distribution for AIX server, or a TME 10 Software Distribution for Windows NT server (“Configuration for a SNA/DS Connection”)
- For an STS connection between a TME 10 Software Distribution for OS/2 server and another TME 10 Software Distribution for OS/2 server, a TME 10 Software Distribution for OS/2 client, or a TME 10 Software Distribution for AIX server (“Configuration for an STS Connection using APPC” on page 116)
- For an APPC connection between an TME 10 Software Distribution for OS/2 client and a TME 10 Software Distribution for OS/2 or TME 10 Software Distribution for AIX server (“Configuration for an APPC Connection on a Client” on page 118) at the client

For information about configuring Personal Communications for communication with NetView DM for MVS, see “Configuring Personal Communications/Communications Server” on page 91.

Configuration for a SNA/DS Connection

This section shows sample output files for connecting a TME 10 Software Distribution for OS/2 server to a TME 10 Software Distribution for OS/2, TME 10 Software Distribution for AIX, or TME 10 Software Distribution for Windows NT server over SNA/DS.

Defining the Local Control Point Profile

```
DEFINE_LOCAL_CP FQ_CP_NAME(ITIBM0PC.PPE0539 )
                DESCRIPTION(Physical Unit 539)
                CP_ALIAS(PPE0539 )
                NAU_ADDRESS(INDEPENDENT_LU)
                NODE_TYPE(EN)
                NODE_ID(X'05D00539')
                NW_FP_SUPPORT(NONE)
                HOST_FP_SUPPORT(YES)
                HOST_FP_LINK_NAME(HOST0001)
                MAX_COMP_LEVEL(NONE)
                MAX_COMP_TOKENS(0);
```

Configuring Personal Communications

Defining the Logical Link Profile

```
DEFINE_LOGICAL_LINK LINK_NAME(LINK0002)
                     FQ_ADJACENT_CP_NAME(ITIBM0PC.PPE0541 )
                     ADJACENT_NODE_TYPE(LEN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X'10005AAFA37304')
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(NO)
                     SOLICIT_SSCP_SESSION(NO)
                     NODE_ID(X'05D00539')
                     ACTIVATE_AT_STARTUP(YES)
                     USE_PUNAME_AS_CPNAME(NO)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Defining the Local LU Profile

```
DEFINE_LOCAL_LU LU_NAME(LT0539A0)
                LU_ALIAS(LT0539A0)
                NAU_ADDRESS(INDEPENDENT_LU);
```

Defining the Partner LU Profile

```
DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(ITIBM0PC.LT0541A0)
                  PARTNER_LU_ALIAS(LT0541A0)
                  PARTNER_LU_UNINTERPRETED_NAME(LT0541A0)
                  MAX_MC_LL_SEND_SIZE(32767)
                  CONV_SECURITY_VERIFICATION(NO)
                  PARALLEL_SESSION_SUPPORT(YES);
```

Defining the Partner LU Location Profile

```
DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(ITIBM0PC.LT0541A0)
                           WILDCARD_ENTRY(NO)
                           FQ_OWNING_CP_NAME(ITIBM0PC.PPE0541 )
                           LOCAL_NODE_NN_SERVER(NO);
```

Defining the Mode Profile

```

DEFINE_MODE  MODE_NAME(LU62      )
              COS_NAME(#CONNECT)
              DEFAULT_RU_SIZE(NO)
              MAX_RU_SIZE_UPPER_BOUND(4096)
              RECEIVE_PACING_WINDOW(4)
              MAX_NEGOTIABLE_SESSION_LIMIT(32767)
              PLU_MODE_SESSION_LIMIT(2)*
              MIN_CONWINNERS_SOURCE(0)
              PACING_TYPE(FIXED)
              COMPRESSION_NEED(PROHIBITED)
              PLU_SLU_COMPRESSION(NONE)
              SLU_PLU_COMPRESSION(NONE);

```

*Note that any value can be inserted here for non-MVS nodes.

Defining the SNA Server TP for Inbound Transmissions

```

DEFINE_TP  SNA_SERVICE_TP_NAME(X'21',008)
           PIP_ALLOWED(NO)
           FILESPEC(I:\SOFTDIST\BIN\FNDTC.EXE)
           CONVERSATION_TYPE(EITHER)
           CONV_SECURITY_RQD(NO)
           SYNC_LEVEL(EITHER)
           TP_OPERATION(QUEUED_AM_STARTED)
           PROGRAM_TYPE(BACKGROUND)
           INCOMING_ALLOCATE_QUEUE_DEPTH(255)
           INCOMING_ALLOCATE_TIMEOUT(INFINITE)
           RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

```

Defining the SNA Server TP for Outbound Transmissions

```

DEFINE_TP  SNA_SERVICE_TP_NAME(X'21',007)
           PIP_ALLOWED(NO)
           FILESPEC(I:\SOFTDIST\BIN\FNDTC.EXE)
           CONVERSATION_TYPE(EITHER)
           CONV_SECURITY_RQD(NO)
           SYNC_LEVEL(EITHER)
           TP_OPERATION(QUEUED_AM_STARTED)
           PROGRAM_TYPE(BACKGROUND)
           INCOMING_ALLOCATE_QUEUE_DEPTH(255)
           INCOMING_ALLOCATE_TIMEOUT(INFINITE)
           RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

```

Starting the Attach Manager

```

START_ATTACH_MANAGER;

```

Configuration for an STS Connection using APPC

If you are connecting a TME 10 Software Distribution for OS/2 server to another TME 10 Software Distribution for OS/2 server, or a TME 10 Software Distribution for AIX server using STS, this sample file configures Personal Communications at both OS/2 servers. (The sample for one server is provided. Specify the same data at the partner server, providing the appropriate values.)

If you are connecting an OS/2 server to a TME 10 Software Distribution for OS/2 client or a TME 10 Software Distribution for AIX server using STS, this sample file configures Personal Communications at the server workstation. A sample file to configure the OS/2 client is given in “Configuration for an APPC Connection on a Client” on page 118.

Defining the Local Control Point Profile

```
DEFINE_LOCAL_CP  FQ_CP_NAME(ITIBM0PC.I9RLW0DZ)
                  CP_ALIAS(I9RLWXXX)
                  NAU_ADDRESS(INDEPENDENT_LU)
                  NODE_TYPE(EN)
                  NODE_ID(X'05D001F7')
                  NW_FP_SUPPORT(NONE)
                  HOST_FP_SUPPORT(YES)
                  HOST_FP_LINK_NAME(HOST0001)
                  MAX_COMP_LEVEL(NONE)
                  MAX_COMP_TOKENS(0);
```

Defining the Logical Link Profile

```
DEFINE_LOGICAL_LINK LINK_NAME(LINK0001)
                     FQ_ADJACENT_CP_NAME(ITIBM0PC.I9RLW06X)
                     ADJACENT_NODE_TYPE(LEARN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X'10005AAFA01504')
                     ETHERNET_FORMAT(NO)
                     CP_CP_SESSION_SUPPORT(NO)
                     ACTIVATE_AT_STARTUP(NO)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     SOLICIT_SSCP_SESSION(NO)
                     MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                     USE_PUNAME_AS_CPNAME(NO)
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Defining the Local Logical Unit

```
DEFINE_LOCAL_LU LU_NAME(I9RLW0ZD)
                LU_ALIAS(I9RLW0ZD)
                NAU_ADDRESS(INDEPENDENT_LU);
```

Defining the Partner Logical Unit

```
DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(ITIBM0PC.I9RLW06X)
                  PARTNER_LU_ALIAS(I9RLW06X)
                  PARTNER_LU_UNINTERPRETED_NAME(I9RLW06X)
                  MAX_MC_LL_SEND_SIZE(32767)
                  CONV_SECURITY_VERIFICATION(NO)
                  PARALLEL_SESSION_SUPPORT(YES);
```

Defining the Partner LU Location Profile

```
DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(ITIBM0PC.I9RLW06X)
                           WILDCARD_ENTRY(NO)
                           FQ_OWNING_CP_NAME(ITIBM0PC.I9RLW06X)
                           LOCAL_NODE_NN_SERVER(NO);
```

Configuration for an APPC Connection on a Client

Defining the Mode Profile

```
DEFINE_MODE  MODE_NAME(LU62      )
              COS_NAME(#CONNECT)
              DEFAULT_RU_SIZE(NO)
              MAX_RU_SIZE_UPPER_BOUND(4096)
              RECEIVE_PACING_WINDOW(63)
              MAX_NEGOTIABLE_SESSION_LIMIT(32767)
              PLU_MODE_SESSION_LIMIT(1)
              MIN_CONWINNERS_SOURCE(0)
              COMPRESSION_NEED(PROHIBITED)
              PLU_SLU_COMPRESSION(NONE)
              SLU_PLU_COMPRESSION(NONE);
```

Defining the Personal Communications Default Values

```
DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                  DEFAULT_MODE_NAME(LU62)
                  MAX_MC_LL_SEND_SIZE(32767)
                  DIRECTORY_FOR_INBOUND_ATTACHES(*)
                  DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                  DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                  DEFAULT_TP_CONV_SECURITY_RQD(NO)
                  MAX_HELD_ALERTS(10);
```

Defining the SNA Server TP for Inbound and Outbound Transmission

```
DEFINE_TP  TP_NAME(NVDM)
            PIP_ALLOWED(NO)
            FILESPEC(C:\SOFTDIST\BIN\FNDSCHD.EXE)
            CONVERSATION_TYPE(ANY_TYPE)
            CONV_SECURITY_RQD(NO)
            SYNC_LEVEL(EITHER)
            TP_OPERATION(QUEUED_AM_STARTED)
            PROGRAM_TYPE(BACKGROUND)
            INCOMING_ALLOCATE_QUEUE_DEPTH(255)
            INCOMING_ALLOCATE_TIMEOUT(INFINITE)
            RECEIVE_ALLOCATE_TIMEOUT(INFINITE);
```

Starting The Attach Manager

```
START_ATTACH_MANAGER;
```

Configuration for an APPC Connection on a Client

This sample file configures Personal Communications at the client workstation.

Defining the Local Control Point Profile

```
DEFINE_LOCAL_CP FQ_CP_NAME(ITIBM0PC.I9RLW06X)

CP_ALIAS(I9RLW06X)
NAU_ADDRESS(INDEPENDENT_LU)
NODE_TYPE(EN)
NODE_ID(X'05D000F9')
NW_FP_SUPPORT(NONE)
HOST_FP_SUPPORT(YES)
HOST_FP_LINK_NAME(HOST0001)
MAX_COMP_LEVEL(NONE)
MAX_COMP_TOKENS(0);
```

Defining the Logical Link Profile

```
DEFINE_LOGICAL_LINK LINK_NAME(HOST0001)
ADJACENT_NODE_TYPE(LEN)
DLC_NAME(IBMTRNET)
ADAPTER_NUMBER(0)
DESTINATION_ADDRESS(X'400010390000')
ETHERNET_FORMAT(NO)
CP_CP_SESSION_SUPPORT(NO)
SOLICIT_SSCP_SESSION(YES)
NODE_ID(X'05D000F9')
ACTIVATE_AT_STARTUP(NO)
USE_PUNAME_AS_CPNAME(NO)
LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
COST_PER_BYTE(USE_ADAPTER_DEFINITION)
SECURITY(USE_ADAPTER_DEFINITION)
PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
USER_DEFINED_1(USE_ADAPTER_DEFINITION)
USER_DEFINED_2(USE_ADAPTER_DEFINITION)
USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOGICAL_LINK LINK_NAME(LINK0002)
FQ_ADJACENT_CP_NAME(ITIBM0PC.I9RLW0DZ)
ADJACENT_NODE_TYPE(LEN)
DLC_NAME(IBMTRNET)
ADAPTER_NUMBER(0)
DESTINATION_ADDRESS(X'10005AAEDCF7')
ETHERNET_FORMAT(NO)
CP_CP_SESSION_SUPPORT(NO)
SOLICIT_SSCP_SESSION(YES)
ACTIVATE_AT_STARTUP(NO)
USE_PUNAME_AS_CPNAME(NO)
LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
```

Configuration for an APPC Connection on a Client

```
LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
COST_PER_BYTE(USE_ADAPTER_DEFINITION)
SECURITY(USE_ADAPTER_DEFINITION)
PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
USER_DEFINED_1(USE_ADAPTER_DEFINITION)
USER_DEFINED_2(USE_ADAPTER_DEFINITION)
USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Defining the Partner Logical Unit Profile

```
DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(ITIBM0PC.I9RLW0DZ)
                   PARTNER_LU_ALIAS(I9RLW0DZ)
                   PARTNER_LU_UNINTERPRETED_NAME(I9RLW0DZ)
                   MAX_MC_LL_SEND_SIZE(32767)
                   CONV_SECURITY_VERIFICATION(NO)
                   PARALLEL_SESSION_SUPPORT(YES);
```

Defining the Partner LU Location Profile

```
DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(ITIBM0PC.I9RLW0DZ)
                             WILDCARD_ENTRY(NO)
                             FQ_OWNING_CP_NAME(ITIBM0PC.I9RLW0DZ)
                             LOCAL_NODE_NN_SERVER(NO);
```

Defining the Mode Profile

```
DEFINE_MODE  MODE_NAME(LU62      )
             COS_NAME(#CONNECT)
             DEFAULT_RU_SIZE(NO)
             MAX_RU_SIZE_UPPER_BOUND(4096)
             RECEIVE_PACING_WINDOW(63)
             MAX_NEGOTIABLE_SESSION_LIMIT(32767)
             PLU_MODE_SESSION_LIMIT(8)4
             MIN_CONWINNERS_SOURCE(0)
             COMPRESSION_NEED(PROHIBITED)
             PLU_SLU_COMPRESSION(NONE)
             SLU_PLU_COMPRESSION(NONE);
```

⁴ It is recommended that the mode profile define a minimum number of sessions of at least 8. This is necessary because multiple processes on the client require separate sessions to communicate with the server.

Defining the Personal Communications Default Values

```
DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                  DEFAULT_MODE_NAME(BLANK)
                  MAX_MC_LL_SEND_SIZE(32767)
                  DIRECTORY_FOR_INBOUND_ATTACHES(*)
                  DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                  DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                  DEFAULT_TP_CONV_SECURITY_RQD(NO)
                  MAX_HELD_ALERTS(10);
```

Defining the SNA Server TP for Inbound and Outbound Transmission

```
DEFINE_TP  TP_NAME(NVDM)
           PIP_ALLOWED(NO)
           FILESPEC(d:\SOFTDIST\BIN\fndcmps.exe)
           PARM_STRING(SNA)
           CONVERSATION_TYPE(EITHER)
           CONV_SECURITY_RQD(NO)
           SYNC_LEVEL(EITHER)
           TP_OPERATION(QUEUED_AM_STARTED)
           PROGRAM_TYPE(FULL_SCREEN)
           INCOMING_ALLOCATE_QUEUE_DEPTH(255)
           INCOMING_ALLOCATE_TIMEOUT(INFINITE)
           RECEIVE_ALLOCATE_TIMEOUT(INFINITE);
```

Starting the Attach Manager

```
START_ATTACH_MANAGER;
```

Chapter 14. Configuring STS Remote Connection Files

This chapter describes how to configure the connection files required to define an STS (Server-to-Server) network using APPC, TCP/IP, IPX, or NetBIOS connections. You need to edit this configuration information when you define or update STS connections to remote nodes.

Editing STS Connection Configuration Files

The configuration information for STS connections is in three text files, which can be edited using a text editor.

The files are:

STS configuration file

This file contains the system parameters for controlling TME 10 Software Distribution, Version 3.1.5 for OS/2's use of the STS network implemented across APPC, TCP/IP, IPX, or NetBIOS connections.

STS connection configuration files

This file contains the details of an APPC, TCP/IP, IPX, or NetBIOS connection to an adjacent node.

Routing table

This file instructs the distribution server which APPC, TCP/IP, IPX, or NetBIOS connection should be used use when distributions are sent to remote targets.

If you edit one of these files while TME 10 Software Distribution, Version 3.1.5 for OS/2 is running, you must use the **rlid** command to reload configuration changes.

STS Configuration File

The STS configuration file, <product_dir>\db\snadscfg, contains a parameter for controlling the use that TME 10 Software Distribution, Version 3.1.5 for OS/2 makes of the STS network, ORIGIN HOP COUNT.

There can be only one configuration file at a distribution server. Your network may include both STS and SNA/DS connections. An SNA/DS connection file can contain other keywords in addition to ORIGIN HOP COUNT (see "SNA/DS Configuration File" on page 133). If you have a mixed network, specify all the keywords in a single file.

You only need to edit this file when performing system tuning, and access is usually restricted to the administrator.

The file is stored with a fixed text format. Enter the keyword once, in uppercase characters, and end it with a colon. Blank and comment lines can be included. Comment lines begin with a number sign (#).

STS Connection Configuration File

Table 20. STS Configuration File Parameters	
Keyword	Description
ORIGIN HOP COUNT	The default hop count to be used for distributions originated by TME 10 Software Distribution, Version 3.1.5 for OS/2 when the hop count is omitted or is specified as zero in the routing table. (See "Editing the STS Routing Table" on page 126.) An appropriate value for this field is 5.

Figure 37 is an example of the STS Configuration File.

STS CONFIGURATION FILE
#
This file should be stored as <product_dir>\db\snadscfg
ORIGIN HOP COUNT: 10

Figure 37. Example of an STS Configuration file

STS Connection Configuration File

STS connection configuration files define the details of APPC, TCP/IP, IPX, or NetBIOS connections to adjacent nodes. Each file is given the name of the connection it defines. Up to 800 connection configuration files can be defined, while up to 100 connections can be active simultaneously.

The files are stored in the directory <product_dir>\db\snadscon, where <product_dir> is the directory where the product was installed. A sample file, called defconft, is provided for STS connections via TCP/IP;.

Access is usually restricted to the administrator. The sample connection configuration file is provided in the <product_dir> directory.

Each file is stored with a fixed text format. Each line starts with one of the keywords described in Table 21 on page 125. Enter each keyword in uppercase and end it with a colon.

Each keyword can be used only once. The order of the keywords is not important, and blank and comment lines can be included. Comment lines begin with a number sign (#).

Table 21. STS Connection Configuration File Parameters	
Keyword	Description
PROTOCOL	Specify either: APPC The APPC protocol is used. TCP/IP The TCP/IP protocol is used. IPX The IPX protocol is used. NBI The NetBIOS protocol is used.
TYPE	STS For an STS (server-to-server) connection type.
REMOTE SERVER NAME	The workstation name of the remote server.

Figure 38 is an example of an STS connection configuration file for the TCP/IP connection protocol. This sample file, called DEFCONF.T, is installed during product installation.

```
# STS CONNECTION CONFIGURATION FILE FOR CONNECTION defconf.t
#
# This connection is used to handle transmissions between
# TME 10 Software Distribution, Version 3.1.5 for OS/2 servers using STS
# across TCP/IP.
#
# This file should be stored as <product_dir>\db\snadscon\defconf.t

PROTOCOL:                TCP/IP
TYPE:                     STS
REMOTE SERVER NAME:       remote_wks_name
```

Figure 38. Example of an STS Connection Configuration File for TCP/IP

Figure 39 on page 126 is an example of an STS connection configuration file for the APPC connection protocol.

Editing the STS Routing Table

```
# STS CONNECTION CONFIGURATION FILE FOR CONNECTION
#
# This connection is used to handle transmissions between
# TME 10 Software Distribution, Version 3.1.5 for OS/2 servers using
# APPC across STS.
#
# This file should be stored as <product_dir>\db\snadscon\defconfa

PROTOCOL:                                APPC
TYPE:                                    STS
REMOTE SERVER NAME:                      sts_server
```

Figure 39. Example of an STS Connection Configuration File for APPC

Editing the STS Routing Table

This section describes how to define the connections in the STS routing table.

The routing table is an editable text file that tells the TME 10 Software Distribution, Version 3.1.5 for OS/2 server which APPC, TCP/IP, IPX, or NetBIOS connection to use when sending a distribution to a remote target. Set up this table to reference the connections that you have configured for your system. A routing table can contain up to 1600 entries.

You do not need to configure a routing table if you do not have remote communication configured. You only need to edit it when you add a new remote connection, or when you want to tune a complex network by careful matching of routes and connections.

A sample routing table is set up for you when you configure remote communication during installation. The file is <product_dir>\db\routetab, where <product_dir> is the directory where the product was installed. You need to change this table to define *routes*.

The routing table contains the definition of the network ID for the distribution server and for each of the distribution clients in a network. It also defines a number of routes. Each route has the following data associated with it:

- Protocol type: APPC, TCP/IP, IPX, NetBIOS, or BOTH (meaning APPC and all other protocol types)
- Addresses of the targets reached by this route
- The name of the connection to use
- The hop count to use on distributions sent on this route.

Planning a Simple Routing Table

In a simple configuration, your distribution server is attached to just one remote node. Only one connection is defined and this takes all distributions to and from the distribution server.

You need only configure a single route. The SNA/DS address should be given as *.* to allow any SNA/DS name to be routed. If included, set the connection service parameters to show that any value is supported on the connection.

Planning a Complex Routing Table

Plan a complex routing table with care. Remember that a distribution is sent on the first route in the table that meets the requirements of the distribution. You should, therefore, put specific SNA/DS addresses before generic ones, and restricted connections before all-purpose ones.

You may find it helpful to draw a diagram of your connections to adjacent NetView DM nodes before you start. In your diagram, you can put the SNA/DS addresses of the remote targets reached through each node, and the services offered by each connection.

The diagram helps you ensure that distributions are routed to the correct destinations, and that no distribution is unexpectedly rejected because there is no connection capable of carrying it.

If you have a busy system, you can reserve some connections for handling small, high-priority distributions. This prevents them from being held up until active, large distributions complete. You can do this by setting the priority and capacity service parameters on the reserved connections.

Determining Destination Addresses

Routes in your routing table must identify the destination target of the link being defined. A target is identified by two values:

- Its domain address
- Its target address

These two values are separated by a period (.) and are expressed in this format in routes in the routing table:

```
<domain address>.<target address>
```

In SNA terminology, these values correspond to:

```
<routing group name>.<routing element name>
```

which is commonly expressed as:

```
<RGN>.<REN>
```

Editing the STS Routing Table

The two values are referred to using different terms on the various platforms that can be included in a TME 10 Software Distribution network. To identify a target's <domain address> and <target address> look for the corresponding terms shown in Table 22.

Table 22. Target Identification Terms		
Type of Target	Domain address	Target address
NetView DM for MVS	Network ID *	LU name *
NetView DM/2	Network ID	LU name
TME 10 Software Distribution Server	domain address	target address
Note: * In a NetView DM for MVS-controlled network, the network ID is the same as the network ID defined for the NetView DM for MVS system it is part of. The LU name uniquely identifies the node within that network. You can find out what this is by contacting the administrator at that site. You set the local network ID just once. This value is picked up for the RGN of all local nodes.		

Defining Routes

Before defining the routes in a routing table, the type of network you are defining must be specified. Specify one of the following values for the NETWORK PROTOCOL keyword at the beginning of the routing table:

- APPC** APPC routes are being defined.
- TCP/IP** TCP/IP routes are being defined. This is the default value, even if this keyword is not included in the routing table.
- BOTH** APPC and TCP/IP are being defined in the routing table.
- NetBIOS** NetBIOS routes are being defined.
- IPX** IPX routes are being defined.

Each route is defined by exactly one line in the routing table. Lines in a routing table can be any length. Blank lines are permitted between routes in the routing table. Comment lines begin with a number sign character (#).

Note: In an STS network, if you do *not* need an SNA/DS connection, do not select either APPC or BOTH (APAR II11831), even if your STS connection uses APPC, to avoid running the Transmission Control Program unnecessarily. Instead, use whichever of the other protocols is used by the clients in the LAN.

Enter the information in Table 23 on page 129, in the order shown, to define a route.

Table 23. Defining Routes in a Routing Table

Parameter	Description
Destination Address	<p>The destination nodes that this route serves.</p> <p>TME 10 Software Distribution, Version 3.1.5 for OS/2 searches the table for an entry matching the destination of each distribution that it sends. If no match is found, a distribution report is generated and returned to the originator of the distribution.</p> <p>Enter the address in the form:</p> <p><domain address>.<target address></p> <p>Each value can be up to eight characters long. The characters must be either uppercase letters, numbers, or the special characters @, #, and \$. For example NETWK1.LU0001.</p> <p>You can use the asterisk (*) and question mark (?) as wildcard characters.</p> <p>See “Determining Destination Addresses” on page 127 for additional information.</p>
Connection Name	<p>The connection name for this route. This name relates to the STS connection configuration file that you create under the <product_dir>\db\snaadcon directory (see “Editing STS Connection Configuration Files” on page 123). For consistency this name can be the host name of the system you are connecting to.</p>
Hop Count	<p>This parameter defines the hop count for all distributions sent out using this route. It indicates the maximum number of nodes that the distribution can legitimately pass through before reaching its destination. The hop count prevents distributions from looping between nodes in the network with contradictory routing tables.</p> <p>Enter the hop count as a decimal digit. Set the value to one if the next node on this route is an end node. If you are unsure of the topology of your network, set the field to 5. Hop count can be omitted from a routing table. If it is not defined, the value defaults to the one defined for Origin Hop Count in the SNA/DS configuration file (see “STS Configuration File” on page 123).</p>

Sample routing tables follow. They are supplied with TME 10 Software Distribution, Version 3.1.5 for OS/2 in the <product_dir>\db\rouetab file. The comment line containing the column headings for the configuration information is included to facilitate reading.

Editing the STS Routing Table

```
# STS ROUTING TABLE
# This table provides STS routing information for
# APPC routes.
# This file should be stored as \<product_dir>\db\routetab

NETWORK PROTOCOL:  APPC
#
# Destination      Connection      Hop
# address          Count
#
SRVSNA1.*          CONNSNA1        5
SRVSNA2.*          CONNSNA2        5
*.*                CONNSNA3        5
```

Figure 40. Example of an STS Routing Table for APPC routes

Figure 41 is an example of an STS routing table for TCP/IP routes.

```
# STS ROUTING TABLE
# This table provides STS routing information for
# TCP/IP routes.
# This file should be stored as \<product_dir>\db\routetab

NETWORK PROTOCOL:  TCP/IP
#
# Destination      Connection      Hop
# Address          Count
#
SRVTCP1.*          CONNTCP1        10
SRVTCP2.*          CONNTCP2        10
*.*                CONNTCP3        10
```

Figure 41. Example of an STS Routing Table for TCP/IP routes

Figure 42 on page 131 is an example of an STS routing table for APPC, TCP/IP, IPX, and/or NetBIOS routes.

# STS ROUTING TABLE		
# This table provides STS routing information for		
# APPC, TCP/IP, IPX, and NetBIOS routes.		
# This file should be stored as \<product_dir>\db\routetab		
NETWORK PROTOCOL: BOTH		
#		
# Destination	Connection	Hop
# Address		Count
#		
SRVSNA1.*	CONNSNA1	10
SRVSNA2.*	CONNSNA2	10
,	CONNSNA3	10
NETWK1.HOST	CONNSNA4	10
SRVTCP1.*	CONNTCP1	10
SRVIPX2.*	CONNIPX2	10
SRVNB13.*	CONNBI3	10

Figure 42. Example of an STS Routing Table for APPC, TCP/IP, IPX, NetBIOS

Defining STS Targets

Target name and address information is specified by way of various parameters. This section is a brief summary of this information in relation to target definition for STS communication. For each target you define:

- **Domain address and target address**

You provide this information when you define a target using the graphical interface or the **addtg** line command. The combination of the two (<domain address>.<target address>) uniquely identifies the target in a network.

- **Protocol type**

Additional target identification information is needed for STS connections. A target's protocol type can be TCP/IP, APPC, IPX, or NetBIOS. Include one of the following in the addtg command, depending on the protocol:

- tp TCP:<hostname>
- tp APPC:<rgn.ren.LU>
- tp IPX:<ip_address>
- tp NBI:<NetBIOS address>

Chapter 15. Configuring SNA/DS Remote Connection Files

This chapter describes how to configure configuration files for an SNA/DS network that uses APPC connections. You need to edit this configuration information to define or update SNA/DS connections to remote nodes.

Editing Connection Configuration Files

The configuration information for SNA/DS connections is held in three different text files, which can be edited using a text editor.

The files are owned by root and can be accessed by the administrator, the same user who can make configuration changes to TME 10 Software Distribution, Version 3.1.5 for OS/2 using the graphical interface.

The files that can be edited are:

SNA/DS configuration file

This file contains the system parameters for controlling TME 10 Software Distribution, Version 3.1.5 for OS/2's use of the SNA/DS network implemented across APPC connections.

SNA/DS connection configuration files

This file contains the details of an APPC connection to an adjacent node.

Routing table

This file instructs the distribution server which APPC connection should be used when distributions are sent to remote targets.

If you edit one of these files while TME 10 Software Distribution, Version 3.1.5 for OS/2 is running, you must use the **rlt** command to reload configuration changes.

SNA/DS Configuration File

The SNA/DS configuration file, <prod_dir>\db\snadscfg where <prod_dir> is the directory where the product was installed, contains system parameters for controlling the use that TME 10 Software Distribution, Version 3.1.5 for OS/2 makes of the SNA/DS network. You only need to edit the SNA/DS configuration file when performing system tuning, and access is usually restricted to the administrator.

The file is stored with a fixed text format. Each line starts with one of the keywords described in Table 24 on page 134. Enter each keyword in uppercase and end it with a colon (:).

Each keyword can be used only once. The order of the keywords is not important, and blank and comment lines can be included. Comment lines begin with a number sign (#).

SNA/DS Configuration File

Table 24. SNA/DS Configuration File Parameters	
Keyword	Description
ORIGIN HOP COUNT	The default hop count to be used for distributions originated by TME 10 Software Distribution, Version 3.1.5 for OS/2 when the hop count is omitted or is specified as zero in the routing table. (See "Editing the SNA/DS Routing Table" on page 137.) An appropriate value for this field is 5.
MAX CRMU	<p>The maximum number of Completion Report Message Units (CRMU) that TME 10 Software Distribution, Version 3.1.5 for OS/2 sends at one time when it detects that Message Unit ID (MU_ID) registries are out of synchronization.</p> <p>You do not normally need to change the value of this field. It is used for performance tuning. If the value is low then resynchronization of registries take longer. If the value is high, then system resources are consumed by the resynchronization and are not available for normal processing.</p> <p>Because resynchronization should be rare, and it is unlikely that a connection would have many MU_IDs outstanding at any time, 5 is a reasonable value.</p>
TRANSMISSION HOLD TIME	<p>The time for which a connection should be held by TME 10 Software Distribution, Version 3.1.5 for OS/2 after a severe transitory error or an error requiring operator intervention has been detected.</p> <p>Enter a value in seconds. An appropriate value is 1000 seconds (16 minutes).</p>
ALLOCATION FAILURE RETRY TIME	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for OS/2 is to wait before attempting to reestablish a conversation on this connection after a previous conversation has been unsuccessful. This prevents thrashing when a connection is unavailable.</p> <p>Enter a decimal number. A reasonable value is 300 seconds (five minutes).</p>

The following is an example of the SNA/DS Configuration File.


```
# SNA/DS CONFIGURATION FILE
#
# This file should be stored as \<prod_dir>\db\snadscfg

ORIGIN HOP COUNT:           10
MAX CRMU:                   5
TRANSMISSION HOLD TIME:     3600
ALLOCATION FAILURE RETRY TIME: 1000
```

Figure 43. Example of an SNA/DS Configuration File

SNA/DS Connection Configuration Files

SNA/DS connection configuration files define the details of the APPC connection to adjacent nodes. Each file is given the name of the connection it defines. Up to 800 connection configuration files can be defined, while up to 100 connections can be active simultaneously.

The files are stored in the directory <prod_dir>\db\snadscon, where <prod_dir> is the directory where the product was installed.

Access is usually restricted to the administrator. Sample connection configuration files are provided in the same directory.

Each file is stored with a fixed text format. Each line starts with one of the keywords described in Table 25. Enter each keyword in uppercase and end it with a colon (:).

Each keyword can be used only once. The order of the keywords is not important, and blank and comment lines can be included. Comment lines begin with a number sign (#).

Table 25 (Page 1 of 2). SNA/DS Connection Configuration File Parameters	
Keyword	Description
FULLY QUALIFIED PARTNER LU NAME	The LU name of the node on the other end of this connection.
LOCAL LU ALIAS NAME	The logical unit alias of the local machine, as specified in the Personal Communications configuration.
REMOTE SERVER NAME	Remote server workstation name
MODE NAME	LU62

|
|

SNA/DS Connection Configuration File

<i>Table 25 (Page 2 of 2). SNA/DS Connection Configuration File Parameters</i>	
Keyword	Description
NEXT DSU	<p>The domain address and target address of the SNA/DS node at the other end of this connection. You must find out this information from the system administrator at the other node.</p> <p>The RGN and REN must be entered as consecutive strings separated by a period.</p>
PROTOCOL	APPC The APPC protocol is used.
RETRY LIMIT	<p>The number of times that TME 10 Software Distribution, Version 3.1.5 for OS/2 is to retry the transmission of a distribution (file or change management command) before reporting an error.</p> <p>Enter a decimal number. A reasonable value is 3.</p>
RECEIVE MU_ID TIME-OUT	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for OS/2 is to wait before retrying the receipt of a distribution. This value prevents thrashing if a transitory error has been detected.</p> <p>Enter a decimal number. A reasonable value is 120.</p>
SEND MU_ID TIME-OUT	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for OS/2 is to wait before retrying the transmission of a distribution. This value prevents thrashing if the remote node has detected a transitory error.</p> <p>Enter a decimal number. A reasonable value is 60.</p>
TRANSMISSION TIME-OUT	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for OS/2 is to wait before retrying the transmission of an MU_ID that failed or was interrupted. This prevents thrashing when a transitory error occurs.</p> <p>Enter a decimal number. A reasonable value is 60.</p>
TYPE	SNA For an SNA/DS connection type.

Figure 44 on page 137 is an example of an SNA/DS connection configuration file for APPC. This sample file, called DEFCONFA, is installed during product installation, under the <prod_dir> directory.

```
# SNA/DS CONNECTION CONFIGURATION FILE FOR CONNECTION DEFCONFA
#
# This connection is used to handle transmissions between
# TME 10 Software Distribution, Version 3.1.5 for OS/2 servers.
#
# This file should be stored as <prod_dir>\db\snadscon\defconfa
PROTOCOL:                APPC
TYPE:                    SNA
FULLY QUALIFIED PARTNER LU NAME: ITIBM0PC.D3C73D01
LOCAL LU ALIAS NAME:      LT0142A0
REMOTE SERVER NAME:       D3C73D01
MODE NAME:                LU62
NEXT DSU:                 ITIBM0PC.D3C73D01
TRANSMISSION TIME-OUT:    60
RETRY LIMIT:              3
SEND MU_ID TIME-OUT:      60
RECEIVE MU_ID TIME-OUT:   120
```

Figure 44. Example of an SNA/DS Connection Configuration File for APPC

Editing the SNA/DS Routing Table

This section describes how to define the connections in the SNA/DS routing table.

The routing table is an editable text file that tells the distribution server which APPC connection to use when sending a distribution to a remote target. Set up this table to reference the connections that you have configured for your system. A routing table can contain up to 1600 entries.

You do not need to configure a routing table if you do not enable remote communication. You only need to edit it when you add a new remote connection, or when you want to tune a complex network by careful matching of routes and connections.

A sample routing table is set up for you when you enable remote communication. The file is <prod_dir>\db\routetab, where <prod_dir> is the directory where the product was installed. You need to change this table to define *routes* in it.

The routing table contains the definition of the network ID for the distribution server and for each of the distribution clients in a network. It also defines a number of routes. Each route has the following data associated with it:

- Protocol type: APPC
- SNA/DS address of the targets reached by this route
- The name of the connection to use
- The SNA/DS hop count to use on distributions sent to this route.

Editing the SNA/DS Routing Table

The following parameters can also be included in a routing table for APPC routes:

- Service parameter thresholds specifying:
 - The minimum distribution priority allowed on the route
 - The maximum size distribution allowed on this route
 - The SNA/DS security provided by this route
 - The SNA/DS protection provided by this route.

Planning a Simple Routing Table

In a simple configuration, your distribution server is attached to just one remote node. Only one connection is defined and this takes all distributions to and from the distribution server.

You need only configure a single route. The SNA/DS address should be given as *.* to allow any SNA/DS name to be routed. If included, set the connection service parameters to show that any value is supported on the connection.

Planning a Complex Routing Table

Plan a complex routing table with care. Remember that a distribution is sent on the first route in the table that meets the requirements of the distribution. You should, therefore, put specific SNA/DS addresses before generic ones, and restricted connections before all-purpose ones.

You may find it helpful to draw a diagram of your connections to adjacent NetView DM nodes before you start. In your diagram, you can put the SNA/DS addresses of the remote targets reached through each node, and the services offered by each connection.

The diagram helps you ensure that distributions are routed to the correct destinations, and that no distribution is unexpectedly rejected because there is no connection capable of carrying it.

If you have a busy system, you can reserve some connections for handling small, high-priority distributions. This prevents them from being held up until active, large distributions complete. You can do this by setting the priority and capacity service parameters on the reserved connections.

Determining Destination Addresses

Routes in your routing table must identify the destination target of the link being defined. A target is identified by two values:

- Its domain address (usually the same as its server name)
- Its target address

These two values are separated by a period (.) and are expressed in this format in routes in the routing table:

`<domain address>.<target address>`

The domain address is the same as a target's server name.

In SNA terminology, these values correspond to:

<routing group name>.<routing element name>

which is commonly expressed as:

<RGN>.<REN>

The two values are referred to using different terms on the various platforms that can be included in a TME 10 Software Distribution, Version 3.1.5 for OS/2 network. To identify a target's <domain address> and <target address> look for these corresponding terms:

<i>Table 26. Target Identification Terms</i>		
Type of Target	Domain address	Target address
NetView DM for MVS	Network ID *	LU name *
TME 10 Software Distribution Server	domain address	target address
Note: * In a NetView DM for MVS-controlled network, the network ID is the same as the network ID defined for the NetView DM for MVS system it is part of. The LU name uniquely identifies the node within that network. You can find out what this is by contacting the administrator at that site. You set the local network ID just once. This value is picked up for the RGN of all local nodes.		

Defining Routes

Before defining the routes in a routing table, the type of network you are defining must be specified. For the NETWORK PROTOCOL keyword, at the beginning of the routing table, specify APPC.

Each route is defined by exactly one line in the routing table. Lines in a routing table can be any length. Blank lines are permitted between routes in the routing table. Comment lines begin with a number sign (#).

Enter information in the order shown in Table 27 on page 140 to define a route.

Editing the SNA/DS Routing Table

Table 27 (Page 1 of 3). Defining a Route in the SNA/DS Routing Table

Parameter	Description
Destination Address	<p>The destination SNA/DS nodes that this route serves, as an SNA RGN.REN address. TME 10 Software Distribution, Version 3.1.5 for OS/2 searches the table for an entry matching the destination of each distribution that it sends. If no match is found, a distribution report is generated and returned to the originator of the distribution.</p> <p>Enter the address in the form:</p> <p><domain address>.<target address></p> <p>Each value can be up to eight characters long. The characters must be either uppercase letters, numbers, or the special characters @, #, and \$. For example NETWK1.LU0001.</p> <p>You can use the asterisk (*) and question mark (?) as wildcard characters.</p> <p>See "Determining Destination Addresses" on page 138 for additional information.</p>
Minimum Distribution Priority	<p>The minimum distribution priority that this route supports. This parameter is optional, and can only be used for APPC routes.</p> <p>When TME 10 Software Distribution, Version 3.1.5 for OS/2 has matched an SNA/DS address, it checks the priority available. If the distribution does not have high enough priority to use this route, TME 10 Software Distribution, Version 3.1.5 for OS/2 continues to search for another one.</p> <p>The priority values are:</p> <p>FAST Allows only distributions with priority FAST on this route.</p> <p>CONTROL Allows only distributions with priority FAST or CONTROL on this route.</p> <p>DATA16 Allows only distributions with priority FAST, CONTROL, or DATA16 on this route.</p> <p>DATA15 Allows any distribution with priority DATA15 or higher on this route.</p> <p>DATA14 ... DATA1 Allow any distribution with priority DATAn or higher on this route, where n is any number in the range 1 through 14.</p> <p>ANY Allows any priority distribution on this route.</p>

Table 27 (Page 2 of 3). Defining a Route in the SNA/DS Routing Table

Parameter	Description
Distribution Protection	<p>The distribution protection that this route provides. The protection parameter specifies whether the distribution must be stored on nonvolatile storage while a DSU has responsibility for it. If a distribution requires more protection than a route can provide, the route is not used. This parameter is optional, and can only be used for APPC routes.</p> <p>The protection values are:</p> <p>LEVEL2 Level-2 protection is provided on this route, which indicates that the distribution is safe-stored in nonvolatile storage.</p> <p>ANY Any requested level of protection is supported on this route.</p>
Maximum Size Distribution	<p>The maximum size distribution that this route supports. This is the capacity of the route. If a distribution is larger than this size, the route is not used. This parameter is optional, and can only be used for APPC routes.</p> <p>The capacity values are:</p> <p>0DATA No distribution carrying data is allowed on this connection. Only reports are sent.</p> <p>1MEGABYTE Only distributions with less than 1 MB of data are sent on this connection.</p> <p>4MEGABYTES Only distributions with less than 4 MB of data are sent on this connection.</p> <p>16MEGABYTES All distributions can use this connection.</p> <p>ANY All distributions can use this connection regardless of the amount of data being carried.</p>

Editing the SNA/DS Routing Table

Table 27 (Page 3 of 3). Defining a Route in the SNA/DS Routing Table	
Parameter	Description
Distribution Security	<p>The distribution security that this route provides. The security parameter specifies that the distribution is to be safeguarded from unauthorized access while it is being sent through the DS network. If a distribution requests higher security than a route can provide, the route is not used. This parameter is optional, and can only be used for APPC routes.</p> <p>The security values are:</p> <p>LEVEL2</p> <p>Level-2 security is provided on this route, which indicates that DS should route the distribution on sessions that are designated secure.</p> <p>ANY</p> <p>Any requested level of security is supported on this route.</p>
Connection Name	<p>The connection name for this route. This name relates to the SNA/DS connection file that you create under <prod_dir>\db\snadscon, where <prod_dir> is the directory where the product was installed. For consistency this name can be the host name of the system you are connecting to.</p>
Hop Count	<p>This parameter defines the hop count for all distributions sent out using this route. It indicates the maximum number of SNA/DS nodes that the distribution can legitimately pass through before reaching its destination. The hop count prevents distributions from looping between SNA/DS nodes in the network with contradictory routing tables.</p> <p>Enter the hop count as a decimal digit. Set the value to one if the next node on this route is an end node. If you are unsure of the topology of your network, set the field to 5. Hop count can be omitted from a routing table. If it is not defined, the value defaults to the one defined for Origin Hop Count in the SNA/DS configuration file (see "SNA/DS Configuration File" on page 133).</p>

Sample routing tables follow. They are supplied with TME 10 Software Distribution, Version 3.1.5 for OS/2 in the <prod_dir>\db\rouetab file, where <prod_dir> is the directory where the product was installed. Figure 45 on page 143 is an example of an SNA/DS routing table for APPC routes.


```
# SNA/DS ROUTING TABLE
# This table provides SNA/DS routing information for a
# TME 10 Software Distribution, Version 3.1.5 for OS/2 SNA network.
# This file should be stored as <prod_dir>\db\rouetatab

NETWORK PROTOCOL:  APPC
#
# Destination  Priority Protection  Capacity  Security  Connection  Hop
# Address
#
SRVSNA1.*      DATA16   ANY       1MEGABYTE ANY       CONNSNA1
SRVSNA2.*      ANY       ANY       ANY       ANY       CONNSNA2
*.*           ANY       ANY       ANY       ANY       CONNSNA3
```

Figure 45. Example of an SNA/DS Routing Table for APPC routes

An example of an SNA/DS routing table for APPC and TCP/IP connection protocols is shown in Figure 46.

```
# SNA/DS ROUTING TABLE
# This table provides SNA/DS routing information for
# a TME 10 Software Distribution for OS/2 combination TCP/IP and APPC network.
# This file should be stored as \<prod_dir>\db\rouetatab

NETWORK PROTOCOL:  BOTH
#
# SNA/DS      Priority Protection  Capacity  Security  Connection  Hop
# Destination
# address
#
ITIBM0PC.CTNDM15G  ANY    ANY    ANY    ANY    CTNDM15G    5
SNA001.SNA001      ANY    ANY    ANY    ANY    POWER       5
SERVER01.SERVER01  ANY    ANY    ANY    ANY    SERVER01
SERVER02.SERVER02  ANY    ANY    ANY    ANY    SERVER02
```

Figure 46. Example of an SNA/DS Routing Table for both APPC and TCP/IP protocols

Part 3. TME 10 Software Distribution Scenarios

This part presents scenarios that guide you through typical tasks using the TME 10 Software Distribution functions. The scenarios show how to:

- Prepare software using generic preparation (“Scenario 1: Preparing a Non-CID-Enabled Application in Basic Mode” on page 152 and “Scenario 3: Preparing a Non-CID-Enabled Application in Advanced Mode” on page 166)
- Prepare software using DiskCamera (“Scenario 4: Replicating an Installation with DiskCamera” on page 181 and “Scenario 5: Installing an Application from the Internet using DiskCamera” on page 194)
- Do a push installation of a software object to clients (“Scenario 2: Installing a Software Object to a Target (Push Installation)” on page 161)
- Authorize a client to do pull installations (“Scenario 6: Defining and Authorizing a Client to Install Software Objects” on page 205)
- Do a pull installation of a software object from a client (“Scenario 7: Installing an Application at a Client (Pull Installation)” on page 211)
- Set up the environment required for CID software preparation (Chapter 17, “Setting Up for CID Preparation” on page 215).

Chapter 16. Software Distribution Scenarios

The following scenarios show the preparation, distribution, and installation of applications.

The Software Distribution Environment

Before running the scenarios, be sure that you have, for example, the environment shown in Figure 47.

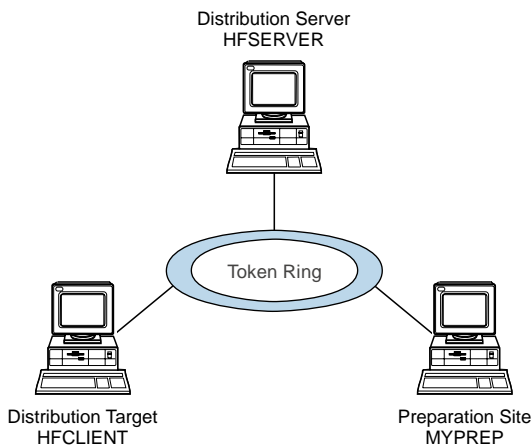


Figure 47. Software distribution environment

Be sure also that you have completed the following activities:

1 Installing the distribution server

The TME 10 Software Distribution server must be installed and configured on the LAN administrator workstation. The system name used during configuration is, for these scenarios, HFSERVER. See Chapter 9, "Installing an OS/2 Distribution Server" on page 53 for a detailed explanation of installing and configuring a TME 10 Software Distribution server.

2 Installing the distribution client

The TME 10 Software Distribution client must be installed and configured on the target workstations. For these scenarios, the system names used during configuration are HFCLIENT and MYPREP. Both HFCLIENT and MYPREP use HFSERVER as their distribution server.

3 Starting the distribution server

The server is automatically started at system startup. If for any reason a distribution server (in this example, HFSERVER) has been stopped, restart it by performing the following steps:

- a From the TME 10 Software Distribution for OS/2 Server window, open the TME 10 Software Distribution for OS/2 Catalog window, as shown in Figure 48 on page 148.

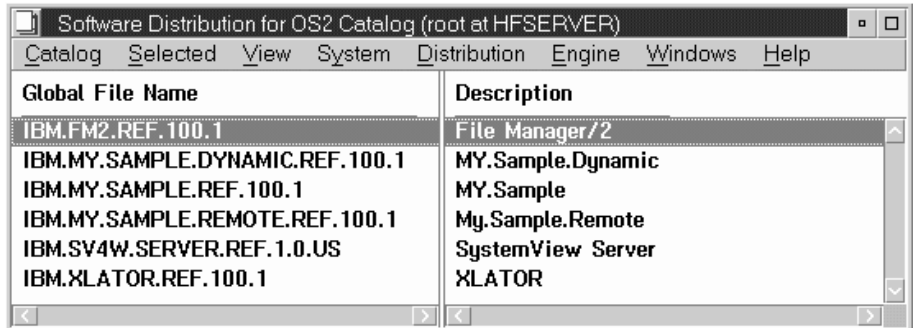


Figure 48. TME 10 Software Distribution for OS/2 Catalog window

- b Select **Engine** from the menu bar.
- c Select **Start the system** from the pull-down menu.

The TME 10 Software Distribution Agent window is displayed, as shown in Figure 49.

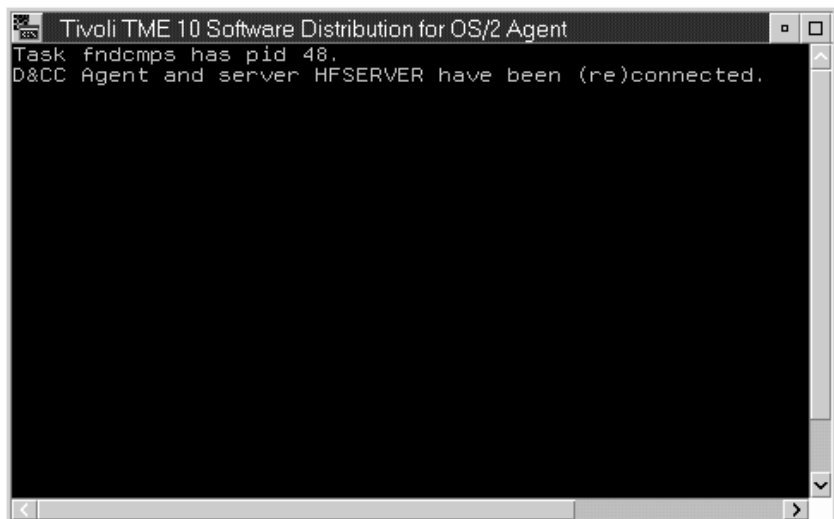


Figure 49. TME 10 Software Distribution Agent window

Wait for the message:

D&CC Agent and server HFSSERVER have been (re)connected.

The Catalog informational message is displayed, as shown in Figure 50.

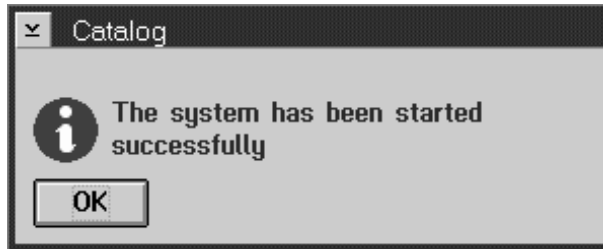


Figure 50. Catalog informational message

- d** Click on the **OK** push button. The restarting of the distribution server is completed.

4 Defining the distribution clients

To define a group of clients as the target for distribution, refer to “Scenario 6: Defining and Authorizing a Client to Install Software Objects” on page 205.

5 Starting the distribution clients

When a distribution server starts, all of its distribution clients that are started are automatically connected to the server. To view the list of the distribution clients, enter the following command from an OS/2 window:

```
nvdm lstg *
```

If for any reason a distribution client (in this example, HFCLIENT) has been stopped, restart it by performing the following steps:

- a** From the TME 10 Software Distribution Client for OS/2 window, open the TME 10 Software Distribution for OS/2 Catalog window, as shown in Figure 51 on page 150.

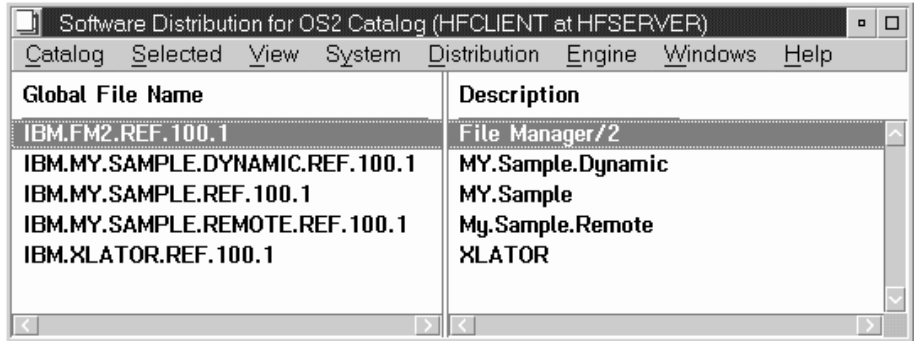


Figure 51. TME 10 Software Distribution for OS/2 Catalog window

- b** Select **Engine** from the menu bar.
- c** Select **Start the system** from the pull-down menu.

The TME 10 Software Distribution Agent window is displayed, as shown in Figure 52.

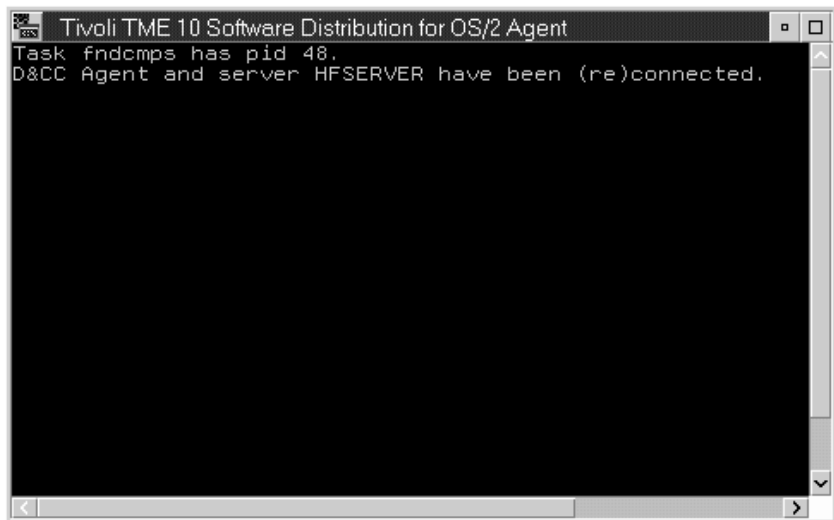


Figure 52. TME 10 Software Distribution Agent window

Wait for the message:

D&CC Agent and server HFSERVER have been (re)connected.

The Catalog informational message is displayed, as shown in Figure 53 on page 151.

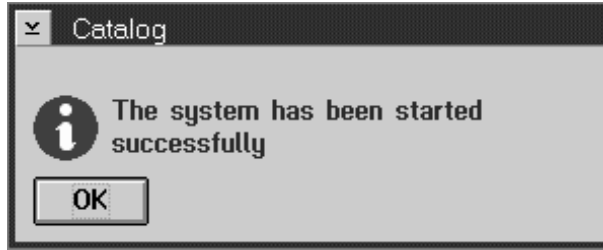


Figure 53. Catalog informational message

- d Click on the **OK** push button. The restarting of the distribution client is completed.

6 Setting Up the Code Server

If you plan to distribute CID-enabled software, set up the code server as explained in “Completing the Setup on the Code Server Workstation” on page 224.

Distribution Database Recovery Tool

This tool is part of the distribution server. The corresponding icon is in the product folder, as shown in Figure 67 on page 161. It provides a recovery procedure that you can run on the distribution server catalog when a specific error occurs.

If, following an error condition, you find this message in the distribution server message log:

```
Open file for <file_name> record returned - 40
```

after making sure all software distribution functions are stopped, click on the TME 10 Software Distribution Database Recovery Tool icon.

Depending on the error condition, the recovery tool either rolls back the catalog to its pre-error state, or updates it correctly.

After the recovery procedure has run successfully, you can restart the distribution server.

TME 10 Software Distribution Database Cleanup Tool

This tool is part of the distribution server. The corresponding icon is in the product folder, as shown in Figure 67 on page 161. It reclaims unused space resulting from deletion of entries from the database.

Before starting this tool, stop the distribution server.

To prevent database corruption, the tool makes a copy of the original database. The copied database tables are renamed with the extension bak and are stored under the database product directory. Therefore you must ensure that enough disk space is available for the backup copy. You can, optionally, manually save the *.bak tables in

Preparing a Non-CID Application (Basic)

your own backup repository. To delete the *.bak copy from the TME 10 Software Distribution product directory, type the `fnddbde1` command from an OS/2 window or full screen. Every backup copy of *.bak files overlays any existing ones.

Scenario 1: Preparing a Non-CID-Enabled Application in Basic Mode

During this scenario, you create a software object and copy it into the catalog at the distribution server.

Before beginning software preparation, perform these steps:

- 1 Make the sample directory on the **F** drive of the preparation site workstation
- 2 Create `sample1.dat` and `sample2.dat` files under the sample directory

To create a software object for a non-CID-enabled application, such as the above sample product, perform the following steps at the preparation site:

- 1 Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon View folder is displayed, as shown in Figure 54.

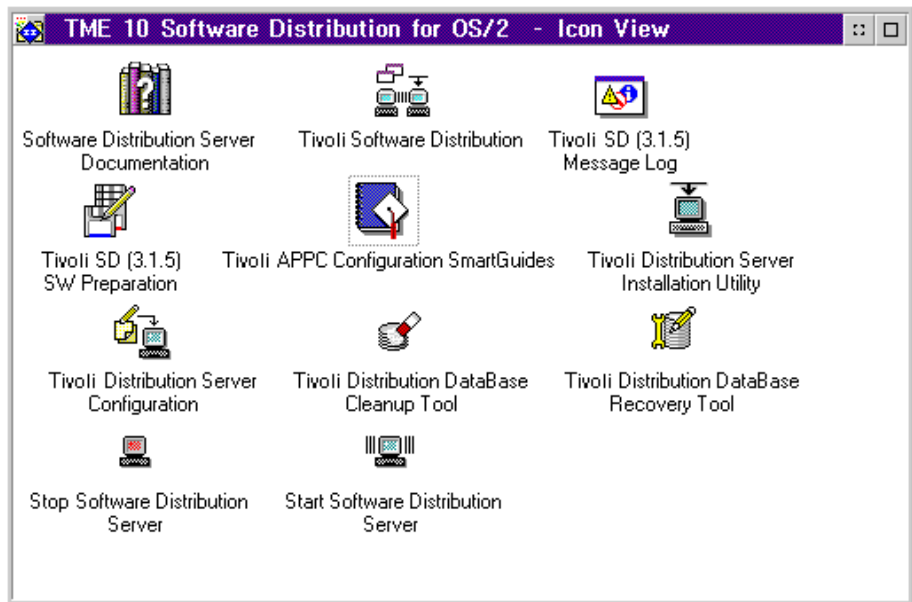


Figure 54. TME 10 Software Distribution for OS/2 - icon view folder

- 2 Double-click on the **Software Preparation** icon.

The Software Preparation folder is displayed, as shown in Figure 55 on page 153.

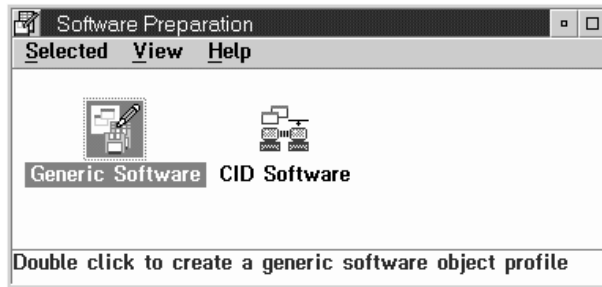


Figure 55. Software preparation folder

- 3 Double-click on the **Generic Software** icon.

The Software Object Profiles window is displayed, as shown in Figure 56.



Figure 56. Software object profiles window

- 4 Select **Software** from the menu bar.
- 5 Select **Create new...** from the **New** choice in the pull-down menu.

The New Software Profile window is displayed, as shown in Figure 57 on page 154.

Preparing a Non-CID Application (Basic)

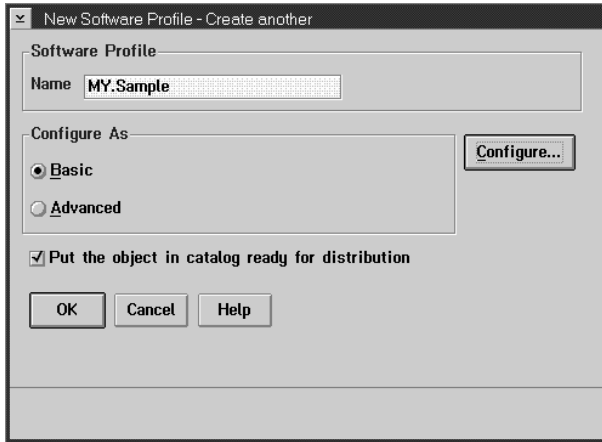


Figure 57. New software profile window

- 6 For **Name**, enter **MY.Sample**.
- 7 Select the **Basic** radio button in the **Configure As** group to a basic creation of a software profile.
- 8 Click on the **Configure...** push button.

The MY.Sample - Basic Configuration notebook is displayed, as shown in Figure 58 on page 155.

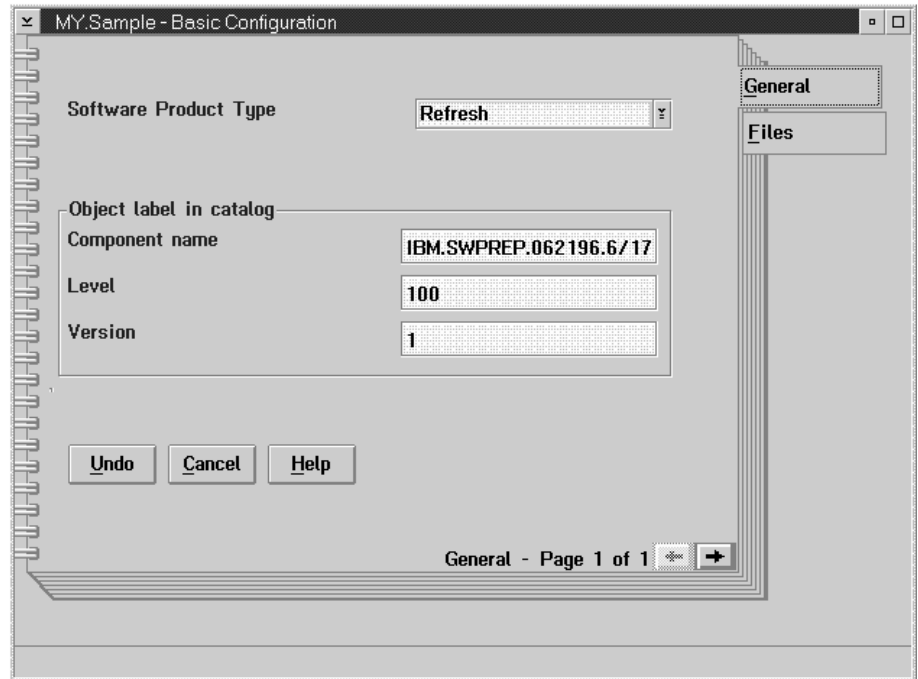


Figure 58. MY.Sample - basic configuration: general page

- 9 Leave the defaults in the General page.
- 10 Select the **Files** tab. The Files page is displayed.

Preparing a Non-CID Application (Basic)



Figure 59. MY.Sample - basic configuration: files page

- 11 In the Files page, choose the files to be included in the software object. Click on the **Find** push button and the Open window is displayed, as shown in Figure 60.

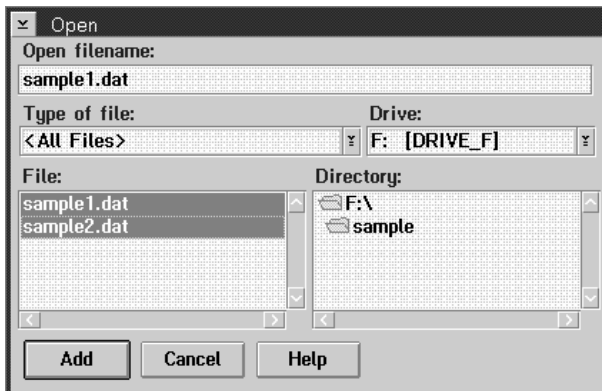


Figure 60. Open window

- 12 Choose the files that will be included in the software object and click on the **Add** push button.

The selected files are included in the Files in software object container on the Files page, as shown in Figure 61 on page 157.

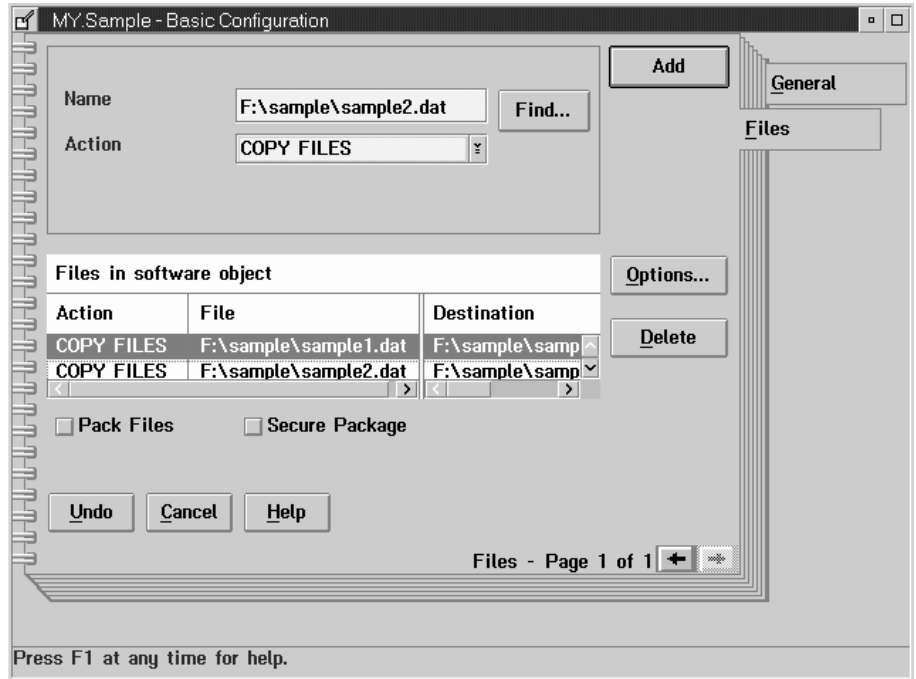


Figure 61. MY.Sample - basic configuration: files page

- 13 You may need to change the destination of the files that will be installed from **F** to **C**. To make this change, select the **sample1.dat** file in the Files in software object container and choose the **Options** push button. The File Options window is displayed, as shown in Figure 62 on page 158.

Preparing a Non-CID Application (Basic)

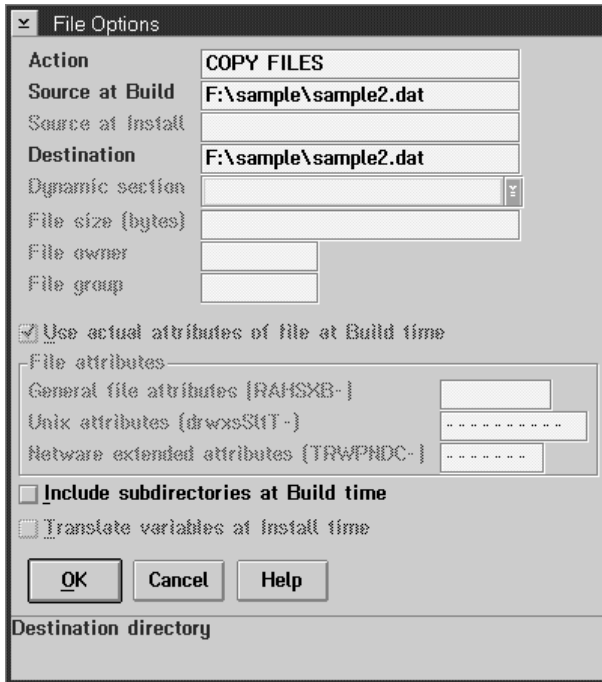


Figure 62. File options window

- 14** Change the destination of **sample1.dat** from **F** to **C**, by changing the installation path in the **Destination** entry field.
- 15** Click on the **OK** push button.
- 16** Repeat the steps 12, 13 and 14 for **sample2.dat** file.
- 17** Close the notebook. The New Software Profile window is displayed again, as shown in Figure 63 on page 159.

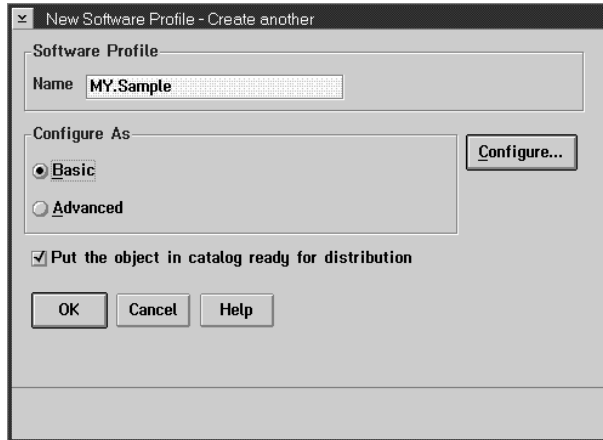


Figure 63. New software profile window

- 18 Leave the check box checked and click on the **OK** push button to build the software object and put it into the catalog.
- 19 A progress indicator is displayed, as shown in Figure 64.

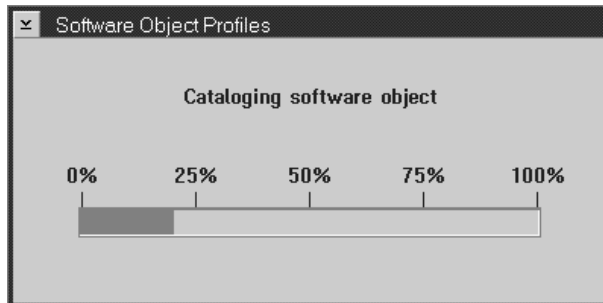


Figure 64. Software object profiles progress indicator

- 20 A Software Object Profiles informational message is displayed, as shown in Figure 65 on page 160.

Preparing a Non-CID Application (Basic)



Figure 65. Software object profiles informational message

- 21 Click on the **OK** push button. The Software Object Profiles window is displayed. It contains the **MY.Sample** software object profile just created, as shown in Figure 66.

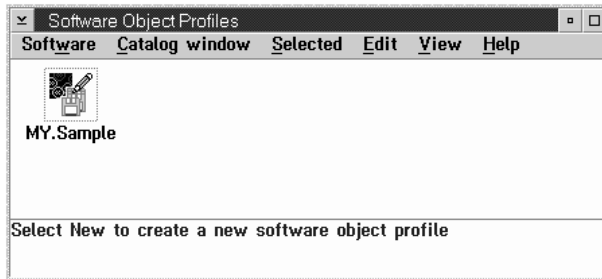


Figure 66. Software objects profile window

Now the **MY.Sample** software object is ready to be installed.

Scenario 2: Installing a Software Object to a Target (Push Installation)

In this scenario, you install a software object from the distribution server catalog to a client workstation.

- 1 Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon View folder is displayed, as shown in Figure 67.

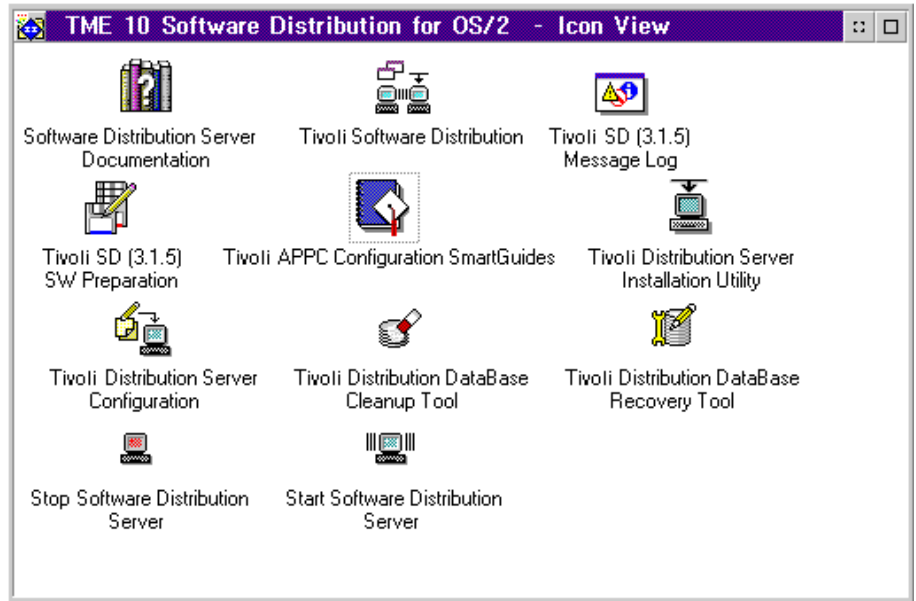


Figure 67. TME 10 Software Distribution for OS/2 - icon view folder

- 2 Double-click on the **TME 10 Software Distribution** icon.

The TME 10 Software Distribution for OS/2 Logon window is displayed, as shown in Figure 68 on page 162.

Installing a Non-CID Application (Push)



Figure 68. TME 10 Software Distribution for OS/2 logon window

- 3 Enter your user ID and password, and click on the **Logon** push button.

The TME 10 Software Distribution for OS/2 Catalog window is displayed, as shown in Figure 69.

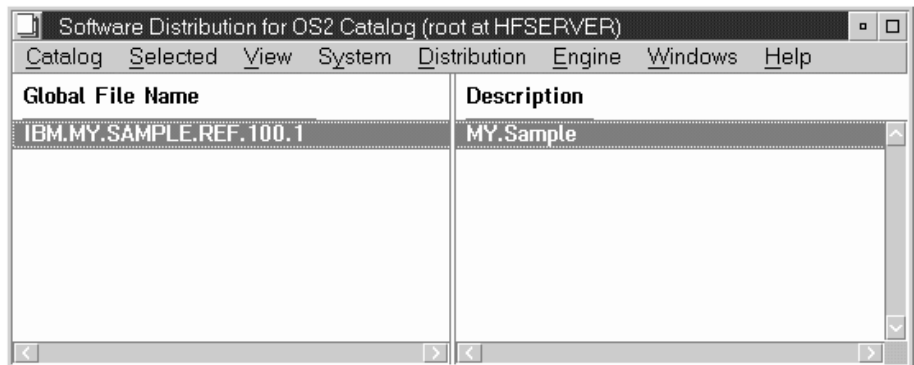


Figure 69. TME 10 Software Distribution for OS/2 Catalog window

Then perform the following steps:

- 4 Select **MY.Sample** software object.
- 5 Select **Install...** from the **Selected** menu choice. The Install Change Files window is displayed, as shown in Figure 70.

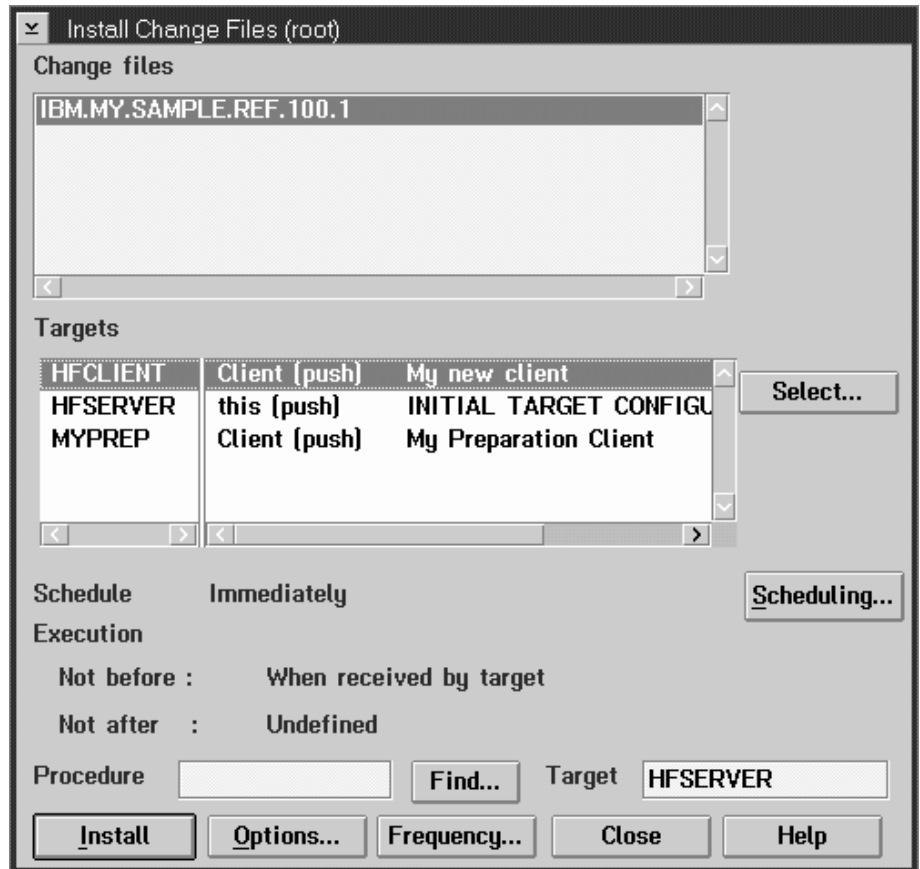


Figure 70. Install Change Files window

- 6 Select the target in the Targets list.
- 7 To schedule the installation, click on the **Scheduling...** push button. The Scheduling Information window is displayed, as shown in Figure 71 on page 164.

⁵ The terms *change file* and *software object* are synonyms.

Installing a Non-CID Application (Push)

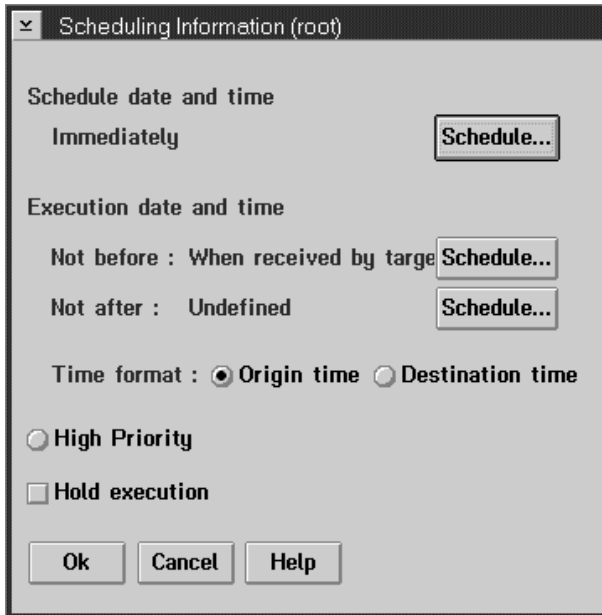


Figure 71. Scheduling information window

- 8 Click on the first **Schedule...** push button. The Schedule Time window is displayed, as shown in Figure 72.

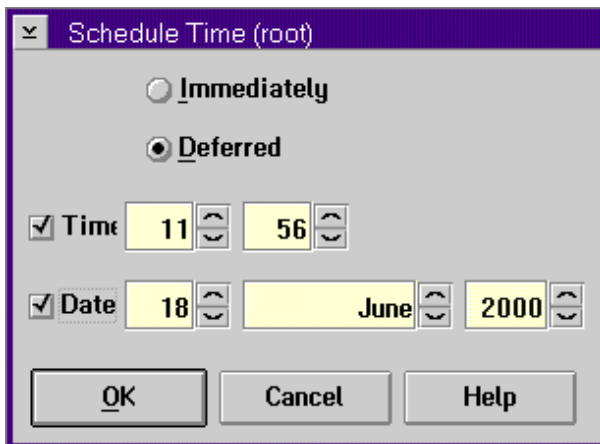


Figure 72. Schedule time window

- 9 Enter the new time and date and click on the **OK** push button. The Scheduling Information window is displayed again, as shown in Figure 73 on page 165.



Figure 73. Scheduling information window

- 10 Click on the **OK** push button. The Install Change Files window is displayed, as shown in Figure 70 on page 163.
- 11 Click on the **Install** push button. The Correlators window is displayed, as shown in Figure 74.



Figure 74. Correlators window

- 12 Click on the **OK** push button. The Install Change Files window is displayed again.
- 13 Click on the **Close** push button. The Catalog window is displayed. The installation of the software object is completed.

Preparing a Non-CID Application (Advanced)

To verify the results of installation, see “Verifying the Results of Installation” on page 166.

Verifying the Results of Installation

You can see the messages from the installation:

- In a file that you can access by entering the command NVDM LOG from an OS/2 window.
- In the TME 10 Software Distribution Agent window.

For additional installation messages, perform the following steps to look at the file FNDLOG under <installation_drive>:\SOFTDIST:

- 1 Open an OS/2 window.
- 2 Enter the commands:
Type <installation_drive>:\SOFTDIST\FNDLOG
>C:\install.msg
- 3 Look at the file install.msg in drive C.

If the FNDLOG file contains the message:

FNDSH220E: The scheduler failed to start a task to process a request

then you may need to increase the number of threads in the THREADS=xxx in CONFIG.SYS.

Scenario 3: Preparing a Non-CID-Enabled Application in Advanced Mode

During this scenario you will perform these tasks:

- Use the Remote Directory function to create a software object and copy it into the catalog at the distribution server.
- Create a dynamic software object and copy it into the catalog at the distribution server.

Creating a Software Object Profile Using the Remote Directory Function

In this scenario, you create a software object to distribute a sample product, as follows:

- Make the sample remote directory on the F drive of a workstation different from the preparation site
- Create the sample1.dat and sample2.dat files under the sample remote directory

To create and catalog the software object, perform the following steps at the preparation site:

- 1 Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon View folder is displayed, as shown in Figure 75 on page 167.

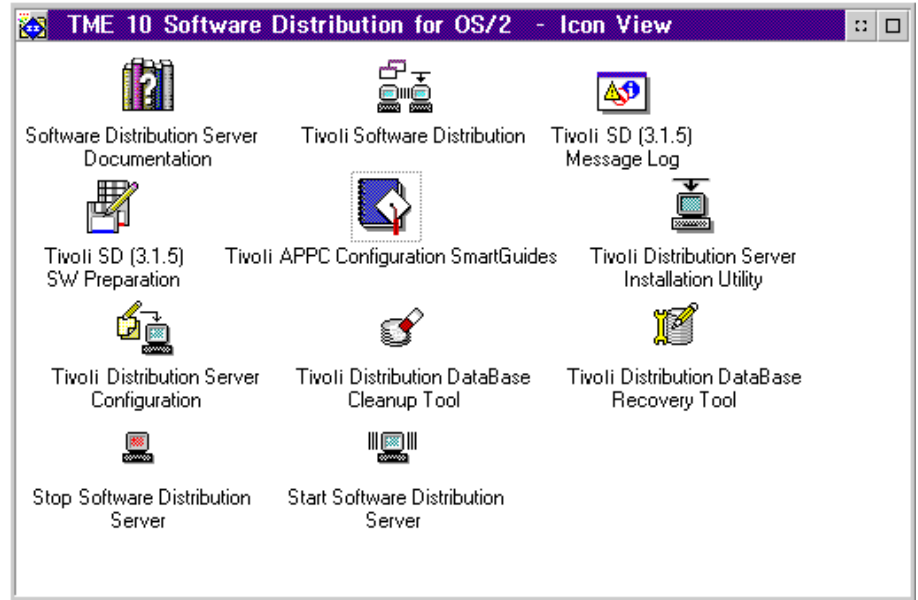


Figure 75. TME 10 Software Distribution for OS/2 - icon view folder

2 Double-click on the **Software Preparation** icon.

The Software Preparation folder is displayed, as shown in Figure 76.

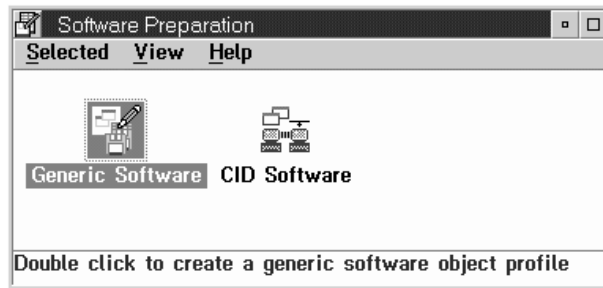


Figure 76. Software preparation folder

3 Double-click on the **Generic Software** icon.

The Software Object Profiles window is displayed, as shown in Figure 77 on page 168.

Preparing a Non-CID Application (Advanced)



Figure 77. Software Object profiles window

- 4 Select **Software** from the menu bar.
 - 5 Select **Create new...** from the **New** choice in the pull-down menu.
- The New Software Profile window is displayed, as shown in Figure 78.

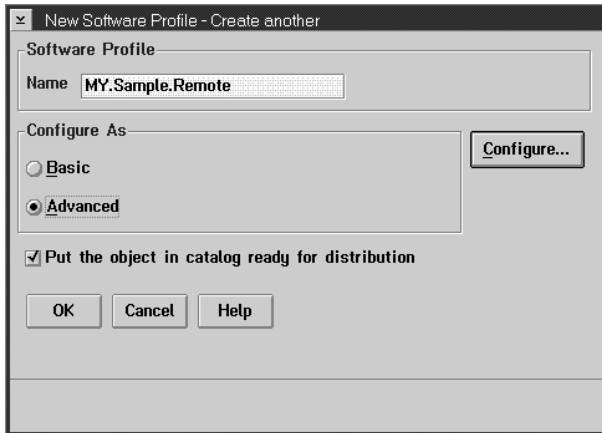


Figure 78. New software profile window

- 6 For **Name**, enter **MY.Sample.Remote**.
- 7 Select the **Advanced** radio button in the **Configure As** group to do an advanced creation of a software profile.
- 8 Click on the **Configure...** push button.

The My.Sample.Remote - Advanced Configuration notebook is displayed, as shown in Figure 79 on page 169.

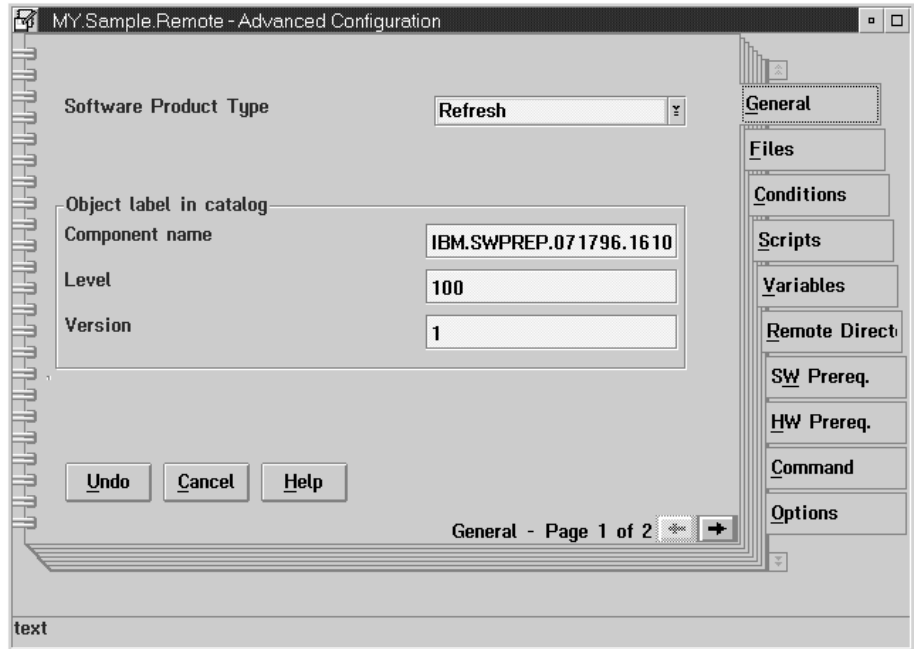


Figure 79. My.Sample.Remote - advanced configuration: general page

- 9 Leave the defaults in the General page.
- 10 Select the **Remote Directory** tab. The Remote Directory page is displayed, as shown in Figure 80 on page 170.

Preparing a Non-CID Application (Advanced)

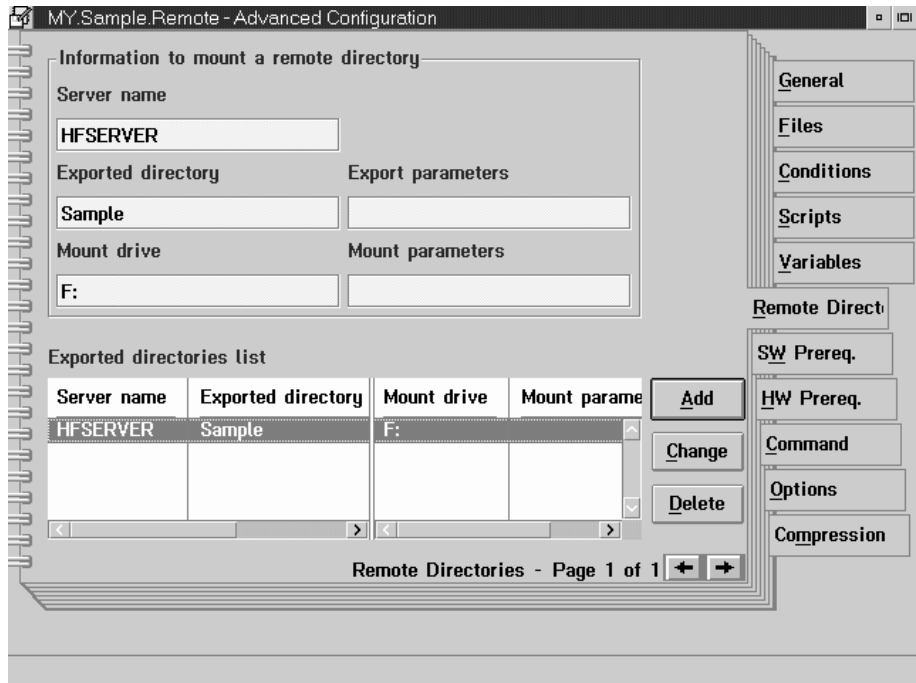


Figure 80. My.Sample.Remote - advanced configuration: remote directory page

In the **Remote Directory** page, choose the remote directory containing the files to be included in the software object. These files will be downloaded using a mount process at installation time.

- 11 Fill in the entry fields contained in the Information to mount a remote directory group, as shown in Figure 80.

Server name	The name of the distribution server.
Exported directory	The remote directory that will be mounted. It contains the files for the software object.
Export parameters	The parameters that define the mode in which the directory is to be exported. This parameter is optional.
Mount drive	The name of the drive from which the destination target is to mount the remote directory.
Mount parameters	The parameters that define the mode in which the drive is to be accessed.

- 12 Click on the **Add** push button.

The mount information is included in the Exported directories list container.

- 13 Close the notebook. The New Software Profile window is displayed again, as shown in Figure 81 on page 171.

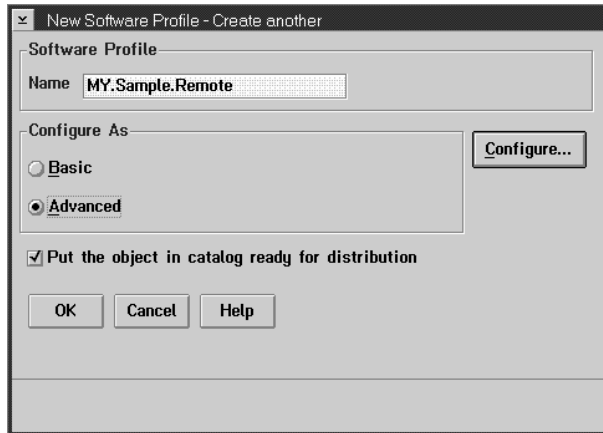


Figure 81. New software profile window

- 14 Leave the check box checked and click on the **OK** push button to build the software object and put it into the catalog.
- 15 A progress indicator is displayed, as shown in Figure 82.

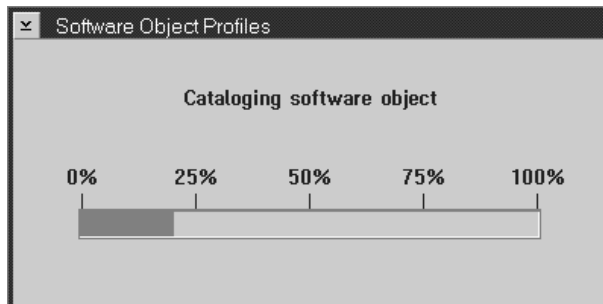


Figure 82. Software object profiles progress indicator

- 16 A Software Object Profiles informational message is displayed, as shown in Figure 83 on page 172.

Preparing a Non-CID Application (Advanced)



Figure 83. Software object profiles informational message

- 17 Click on the **OK** push button, the Software Object Profiles window is displayed. It contains the **My.Sample.Remote** object just created, as shown in Figure 84.

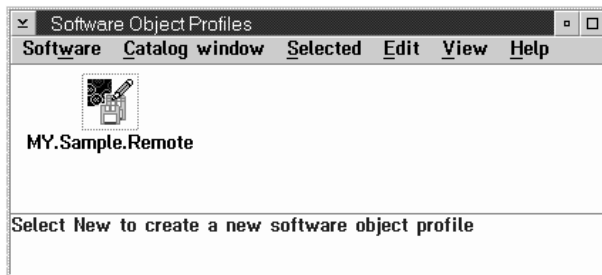


Figure 84. Software objects profile window

Now this software object is ready to be installed. For the installation procedure see "Scenario 2: Installing a Software Object to a Target (Push Installation)" on page 161.

Creating a Dynamic Software Object

A dynamic software object allows you to install different objects (files) on different workstations (targets) during the same installation. Each file is linked to a specific dynamic section containing the installation condition.

In this scenario, you create a dynamic software object that allows dynamic installation. File **sample1.dat** is installed on a target that has a 150-MB hard disk, and **sample2.dat** is installed on a target that has a 200-MB hard disk, with the same install operation.

This example uses the sample product MY.Sample created at the beginning of "Scenario 1: Preparing a Non-CID-Enabled Application in Basic Mode" on page 152.

- 1 Create the **MY.Sample.Dynamic** software profile in advanced mode, as shown in "Creating a Software Object Profile Using the Remote Directory Function" on page 166.
- 2 Open the My.Sample.Dynamic - Advanced Configuration notebook, as shown in Figure 85 on page 173.

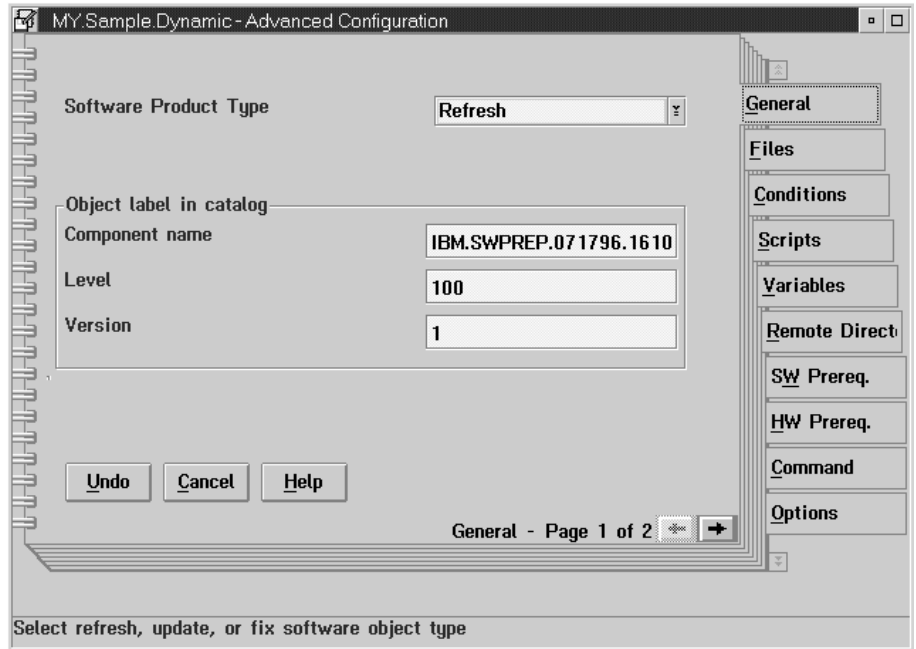


Figure 85. *My.Sample.Dynamic* - advanced configuration: general page

- 3 Leave the defaults in the General page.
- 4 Select the **Files** tab. The Files page is displayed, as shown in Figure 86 on page 174.

Preparing a Non-CID Application (Advanced)



Figure 86. My.Sample.Dynamic - advanced configuration: files page

Now you create two dynamic sections, the first one to install the **sample1.dat** file on a workstation with a disk space of 150 MB, the second to install the **sample2.dat** file on a workstation with a disk space of 200 MB.

- 5 Enter **sample1.dat** in the Name field.
- 6 Leave the default in the Action list.
- 7 Click on the **Open...** push button. The Dynamic Sections - Definition notebook is displayed, as shown in Figure 87 on page 175.

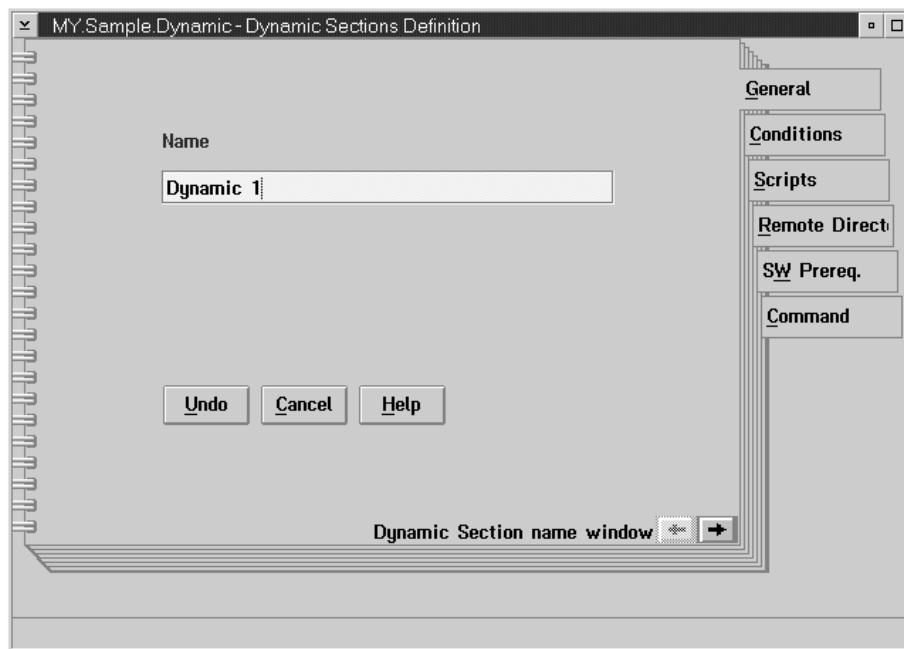


Figure 87. Dynamic sections definition notebook: general page

- 8 For **Name**, enter **Dynamic 1**.
- 9 Click on the **Conditions** tab. The Conditions page is displayed, as shown in Figure 88 on page 176.

Preparing a Non-CID Application (Advanced)

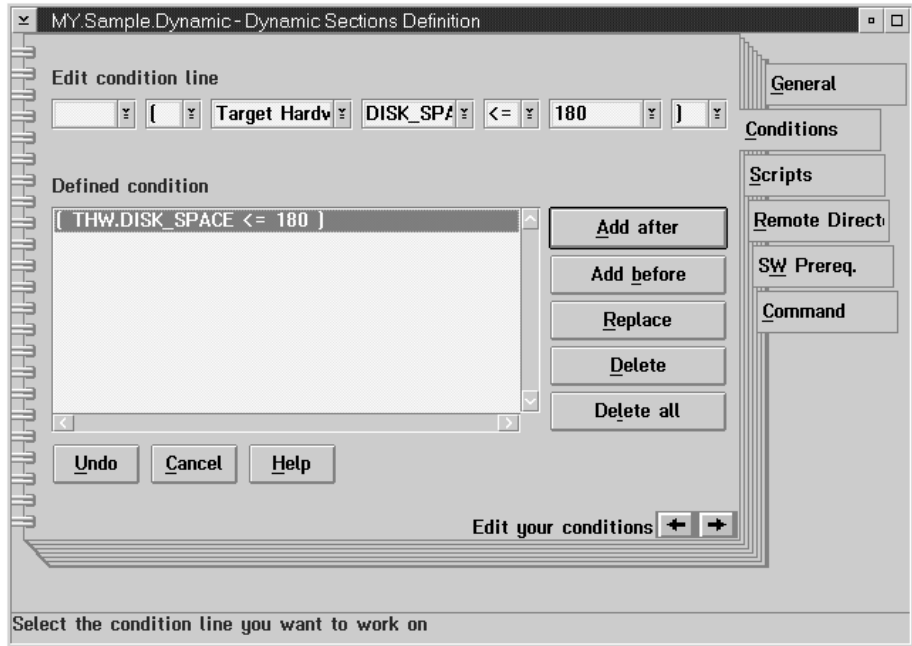


Figure 88. Dynamic sections definition notebook: conditions page

- 10 Fill in the entry fields contained in the Edit condition line group to define the first condition (DRIVE_SIZE < 180M), as shown in Figure 88.
- 11 Click on the **Add after** push button.
The condition is displayed in the Defined condition list.
- 12 Close the notebook. The MY.Sample.Dynamic - Advanced Configuration notebook is displayed again.
- 13 Click on the **Add** push button.
The **sample1.dat** file and the corresponding **Dynamic 1** dynamic section are displayed in the Files in software object container.
- 14 Enter **sample2.dat** in the Name field.
- 15 Leave the default in the Action list.
- 16 Click again on the **Open...** push button. The MY.Sample.Dynamic - Dynamic Sections - Definition notebook is displayed, as shown in Figure 89 on page 177.

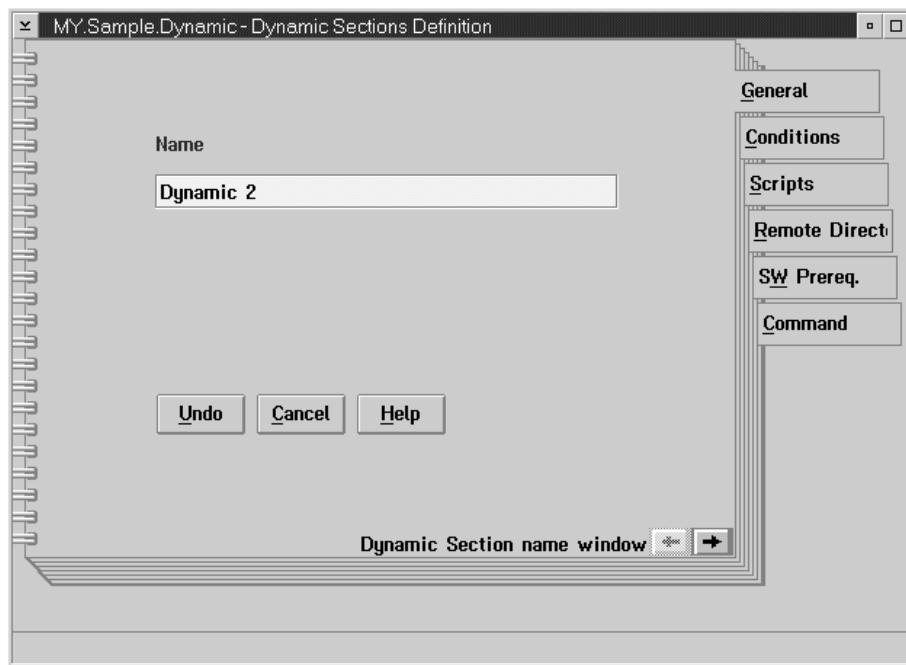


Figure 89. Dynamic sections definition notebook: general page

- 17** For **Name**, enter **Dynamic 2**.
- 18** Click on the **Conditions** tab. The Conditions page is displayed, as shown in Figure 90 on page 178.

Preparing a Non-CID Application (Advanced)

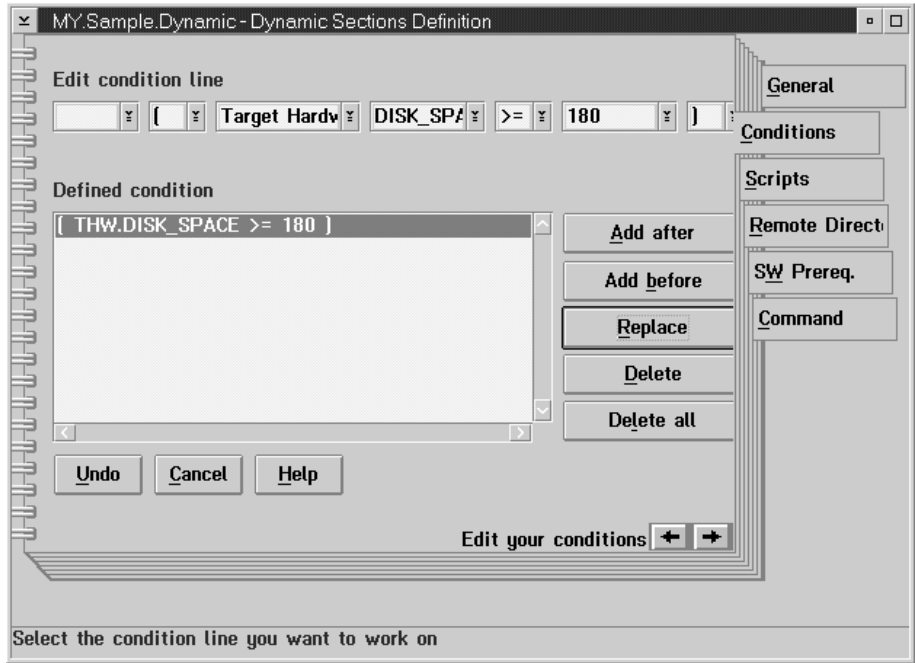


Figure 90. Dynamic sections definition notebook: conditions page

- 19 Fill in the entry fields contained in the Edit condition line group, to define the second condition (DRIVE_SIZE > 180M), as shown in Figure 90.
- 20 Click on the **Add after** push button.
The condition is displayed in the Defined condition list.
- 21 Close the notebook. The My.Sample.Dynamic - Advanced Configuration notebook is displayed again.
- 22 Click on the **Add** push button.
The **sample2.dat** file and the corresponding **Dynamic 2** dynamic section are displayed in the Files in software object container, as shown in Figure 91 on page 179.



Figure 91. MY.Sample.Dynamic - advanced configuration: files page

- 23 Close the notebook. The New Software Profile window is displayed again, as shown in Figure 92.

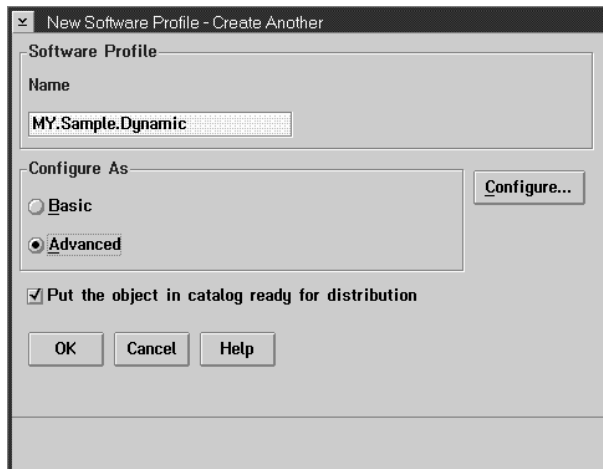


Figure 92. New software profile window

- 24 Leave the check box checked and click on the **OK** push button to build the software object and put it into the Catalog.

Preparing a Non-CID Application (Advanced)

- 25** A progress indicator is displayed, as shown in Figure 93 on page 180.

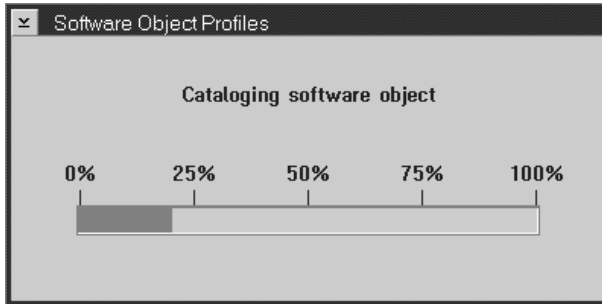


Figure 93. Software object profiles progress indicator

- 26** A Software Object Profiles informational message is displayed, as shown in Figure 94.



Figure 94. Software object profiles informational message

- 27** Click on the **OK** push button, the Software Object Profiles window is displayed, containing the **My.Sample.Dynamic** object just created, as shown in Figure 95.



Figure 95. Software object profiles window

Now this dynamic software object is ready to be installed.

For the installation procedure see “Scenario 2: Installing a Software Object to a Target (Push Installation)” on page 161.

Scenario 4: Replicating an Installation with DiskCamera

This scenario uses an example program called XLATOR, which is an application to be made available to other users. XLATOR does not have an installation program; installing it consists of copying the files in a directory and some subdirectories, and modifying the CONFIG.SYS. DiskCamera makes it easy to replicate this type of installation on multiple workstations.

Using DiskCamera, you can:

- 1 Take a picture of the drive and of selected system files⁶ and workplace objects⁷ *before* and *after* you install an application manually on the workstation where you are doing software preparation
- 2 Using the “before” and “after” pictures, create a software object, which you can catalog at the distribution server. Installing this software object on other workstations has the same effect as the manual installation.

To use DiskCamera to install a software object you must install the operating system in the same directory path on both the preparation site and the target on which you want to install the software object.

If one of the three steps that you perform using DiskCamera fails you must delete the \BIN\DSKCAM.TMP file before using DiskCamera again.

You cannot use DiskCamera to install the WIN-OS/2 applications.

Taking a “Before” Picture

You first take a “before” picture of the E drive, install XLATOR there, and then take an “after” picture. Set aside the E drive for DiskCamera use only, to be sure that the only changes DiskCamera records are those resulting from the installation. For performance

⁶ On OS/2, the default list of files monitored by DiskCamera is OS2INIT.CMD, OS2SYS.INI, OS2.INI, STARTUP.CMD, and CONFIG.SYS. On all Windows clients, CONFIG.SYS, AUTOEXEC.BAT, and all .INI files are monitored. On Windows 95, Windows 98, Windows NT and Windows 2000 clients the Registry is also monitored. You can change the list of files to be monitored (except for the Registry) by editing the file DSKCAM.INI in the TME 10 Software Distribution directory, \BIN subdirectory.

⁷ On OS/2, each workplace object (icon, folder, launch pad, etc.) on the current desktop is monitored, provided it was created with an object ID. DiskCamera does not monitor workplace objects created without an object ID.

On Windows 95 and Windows 98, DiskCamera cannot monitor direct creation of an icon on the desktop or in the Programs folder. To bypass this limitation, after you take the “before” picture, follow these steps:

- a. Install the product
- b. From **Start**, open the Programs folder and create a new folder with the product name
- c. Under this folder, create a new icon and link it to the related product executable file
- d. Then take the “after” picture.

Replicating an Installation with DiskCamera

reasons, it is important that the drive used for DiskCamera contains as few files as possible.

To take the “before” picture, perform the following steps at the preparation site:

- 1 Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon folder is displayed, as shown in Figure 96.

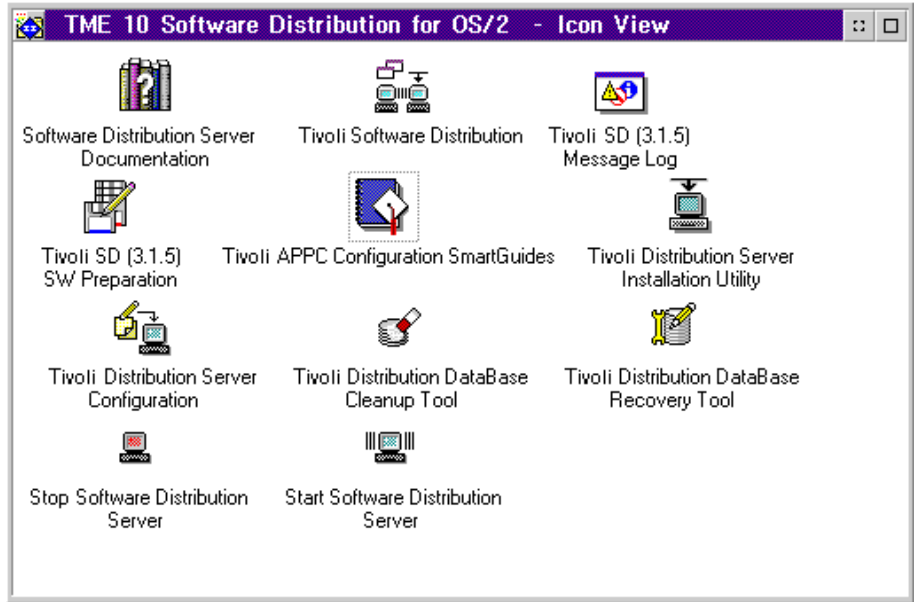


Figure 96. TME 10 Software Distribution for OS/2 - icon view folder

- 2 Double-click on the **Software Preparation** icon.

The Software Preparation folder is displayed, as shown in Figure 97.

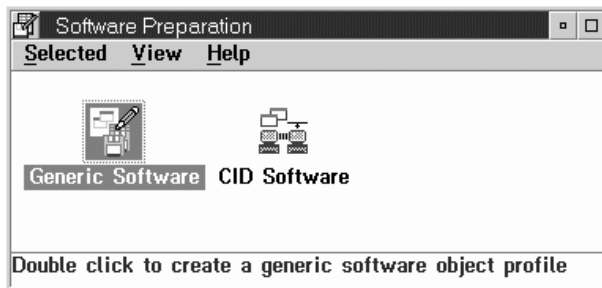


Figure 97. Software preparation folder

- 3 Double-click on the **Generic Software** icon. The Software Object Profiles window is displayed, as shown in Figure 98 on page 183.



Figure 98. Software object profiles window

- 4 Select **Software** from the menu bar.
 - 5 Select **Using DiskCamera...** from the **New** choice in the pull-down menu.
- The Create a software object profile using DiskCamera window is displayed, as shown in Figure 99.



Figure 99. Create a software object profile using DiskCamera window

- 6 Click on the first button (it is the only one available).
- The Take a picture of a disk drive window is displayed, as shown in Figure 100 on page 184.

Replicating an Installation with DiskCamera



Figure 100. Take a picture of a disk drive window

- 7 Select the **E** drive and click on **OK** push button. The first picture starts.

A full-screen window is displayed showing which files are being monitored. This operation could take several minutes.

- 8 An informational message informs you when the disk picture has been successfully completed. Click on the **OK** push button. The Create a software object profile using DiskCamera window is displayed again.

Note that the first button of this window is now checked with a red check mark to indicate that the first step of the DiskCamera process has been completed.

Now you must install the new application at the preparation site.

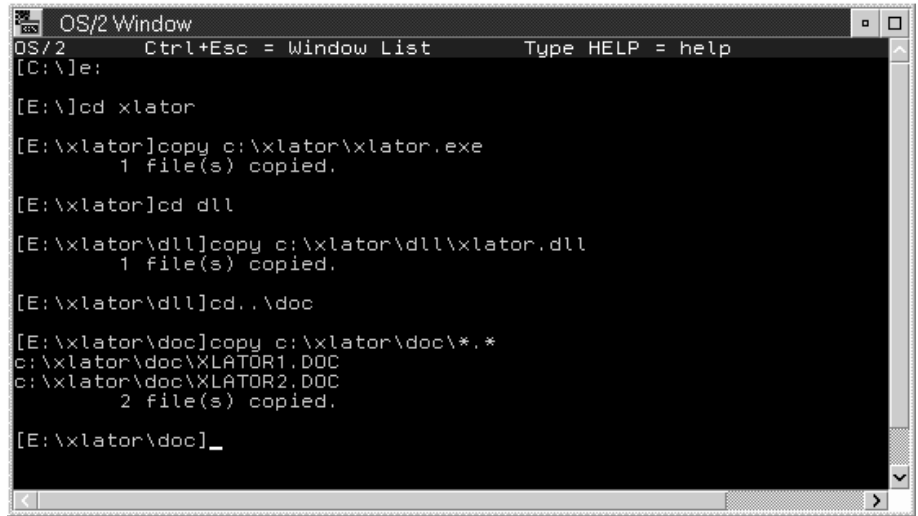
Installing the Application at the Preparation Site

XLATOR is stored on the C drive, under a directory named XLATOR. The XLATOR directory contains the following files:

```
XLATOR\XLATOR.EXE
XLATOR\DLL\XLATOR.DLL
XLATOR\DOC\XLATOR1.DOC
XLATOR\DOC\XLATOR2.DOC
```

To install the application on the E drive, follow the following steps:

- 1 Make a directory XLATOR, with subdirectories DLL and DOC, on the E drive.
- 2 From an OS/2 prompt, copy the files to the E drive, as shown in Figure 101 on page 185.



```

OS/2 Window
OS/2      Ctrl+Esc = Window List      Type HELP = help
[C:\>]e:

[E:\>]cd xlator

[E:\xlator]copy c:\xlator\xlator.exe
          1 file(s) copied.

[E:\xlator]cd dll

[E:\xlator\dll]copy c:\xlator\dll\xlator.dll
          1 file(s) copied.

[E:\xlator\dll]cd ..\doc

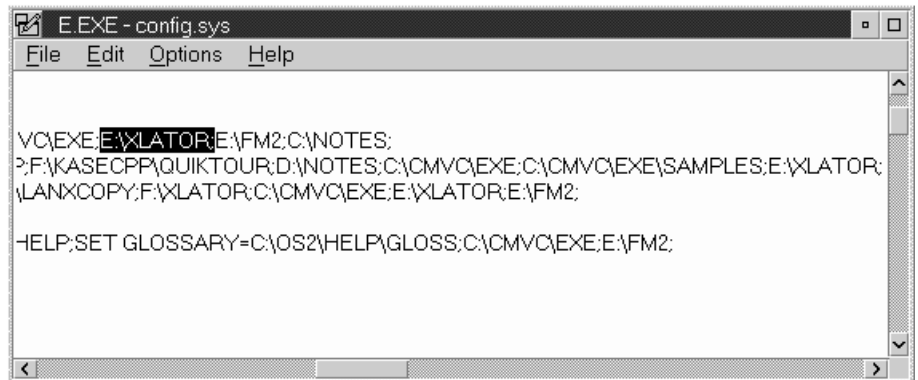
[E:\xlator\doc]copy c:\xlator\doc\*. *
c:\xlator\doc\XLATOR1.DOC
c:\xlator\doc\XLATOR2.DOC
          2 file(s) copied.

[E:\xlator\doc]>_

```

Figure 101. Copying files to the DiskCamera drive

- 3 Using the system editor, edit CONFIG.SYS to add XLATOR to PATH, LIBPATH and DPATH, as shown in Figure 102, and save the changes.



```

E.EXE - config.sys
File Edit Options Help

VC\EXE;E:\XLATOR;E\FM2\C\NOTES;
?;F:\KASEC\PP\QUIKT\OUR;D\NOTES\C\CMVC\EXE;C\CMVC\EXE\SAMPLES;E\XLATOR;
\LAN\XCOPY;F\XLATOR;C\CMVC\EXE;E\XLATOR;E\FM2;

HELP;SET GLOSSARY=C\OS2\HELP\GLOSS;C\CMVC\EXE;E\FM2;

```

Figure 102. Editing CONFIG.SYS

Now you are ready to take the “after” picture.

Taking an “After” Picture

After the installation is completed, the next step is to take the “after” picture of the installation drive and the files being monitored. Note that it is important to be sure the installation is complete before taking the “after” picture; if you need to change the installation, you will have to restart the DiskCamera process from the beginning.

Replicating an Installation with DiskCamera

It is also important to be sure all newly created and changed files are as you want them, and that no other processes running on your workstation have made unwanted changes.

To take the “after” picture of the E drive and of several system files, follow the following steps:

- 1 In the Create a software object profile using DiskCamera window (shown in Figure 99 on page 183) click on the second button to notify DiskCamera that the installation has been completed. A confirmation message is displayed. Click on **OK** to continue.

Note that the second button of this window is now checked with a red check mark to indicate that the second step of the DiskCamera process has been completed.

- 2 In the Create a software object profile using DiskCamera window (shown in Figure 99 on page 183) click on the third button to generate the software object profile.
- 3 The Software Object Profile Generator window is displayed.
Fill in the fields of this window, as shown in Figure 103.

Software Object Profile Generator

Software Object Profile

Name

Object label in catalog

Component name

Level

Version

☒ Insert object in catalog, from where it can be distributed

Check to automatically insert the software object in the catalog

Figure 103. Software Object Profile generator window

Name	The name to appear on the object's icon in the Software Object Profile and Catalog windows. This is also the first part of the software object's name in the software distribution catalog.
Component name	The name of a specific component of a product
Level	A numeric field, normally the maintenance level of the product. It is the sixth part of the software object's name in the software distribution catalog.
Version	A numeric field, normally the version of the product. It is the seventh part of the software object's name in the software distribution catalog.

To catalog the software object, check the Insert object in catalog check box (the default is "checked").

In this example you may want only to create the software object profile and catalog it later so remove the check mark from the check box.

4 Click on **OK**.

DiskCamera takes the "after" picture of the E drive and of the system files. It compares the "before" and "after" pictures and uses the differences to build a software object profile for XLATOR.

A full-screen window is displayed showing which files have changed. This operation could take several minutes.

A progress indicator is displayed, as shown in Figure 82 on page 171.

An informational message is displayed, as shown in Figure 104.

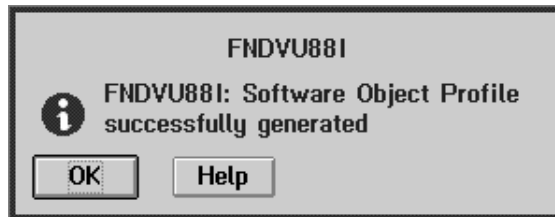


Figure 104. Informational message

5 Click on the **OK** push button. The Create a software object profile using DiskCamera window is displayed, as shown in Figure 105 on page 188.

Replicating an Installation with DiskCamera

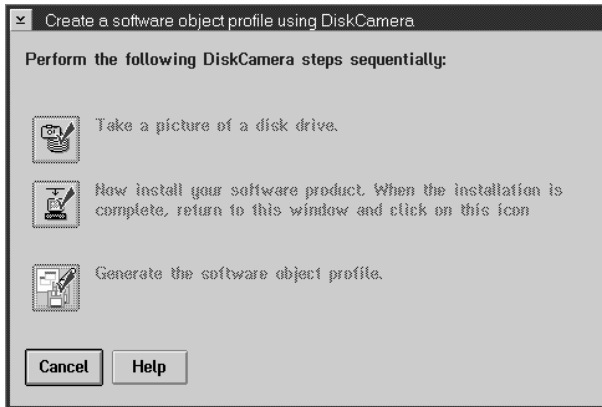


Figure 105. Create a software object profile using DiskCamera window

Now all the buttons are checked. This means that the DiskCamera process has been completed.

6 Click on **Cancel**.

The creation of the XLATOR software object profile has been completed.

The icon for XLATOR is displayed in the Software Object Profiles window, as shown in Figure 106.

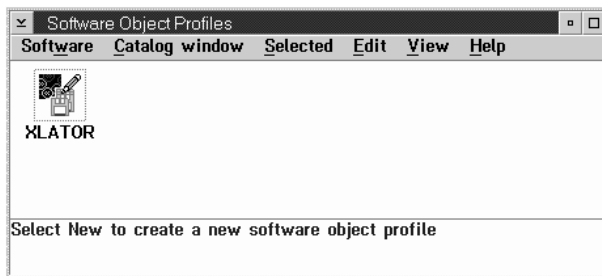


Figure 106. Software object profiles window with XLATOR added

Now you will check the files associated to the XLATOR software object profile. Furthermore you will change the drive where the XLATOR software object will be installed.

7 Double-click on the XLATOR icon in the Software Object Profiles window. The XLATOR - Advanced Configuration notebook is displayed, as shown in Figure 107 on page 189.

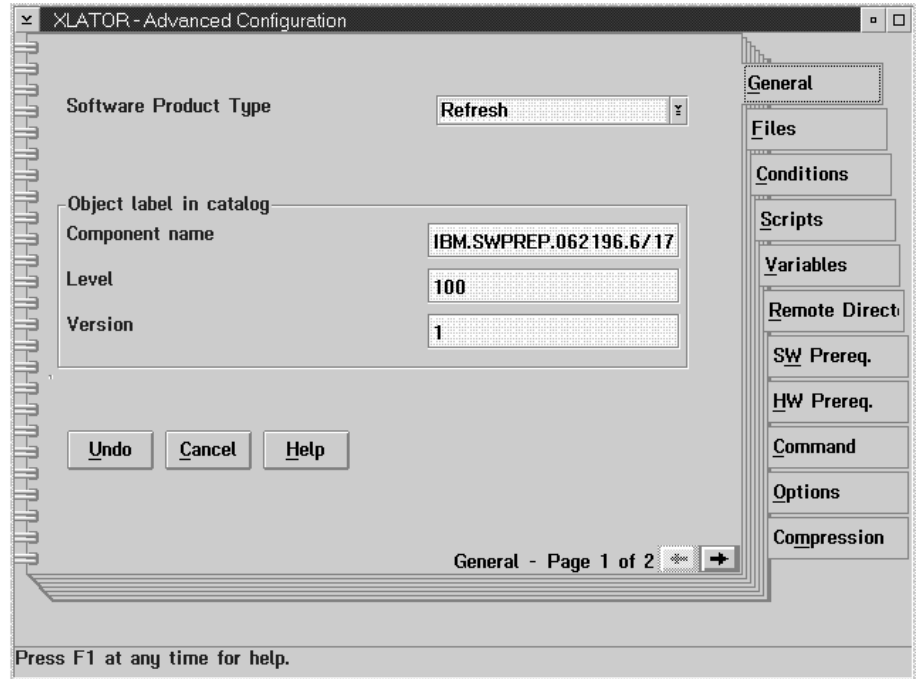


Figure 107. XLATOR - advanced configuration notebook

- 8 Click on the **Files** tab to see the list of files that have been included in the software object. The list is displayed in the Files in software object container, as shown in Figure 108 on page 190. In this page it is possible to add, update, or delete files in the software object.

Replicating an Installation with DiskCamera

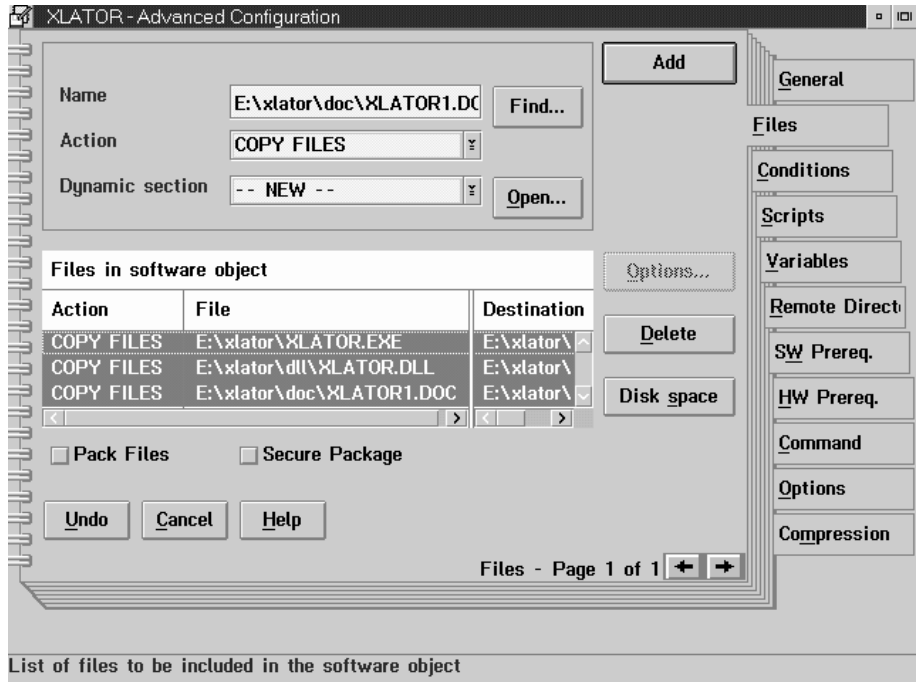


Figure 108. XLATOR - advanced configuration notebook: files page

- 9 To check the variables associated with the software object, click on the **Variables** tab. The **Variables** page is displayed, as shown in Figure 109 on page 191.

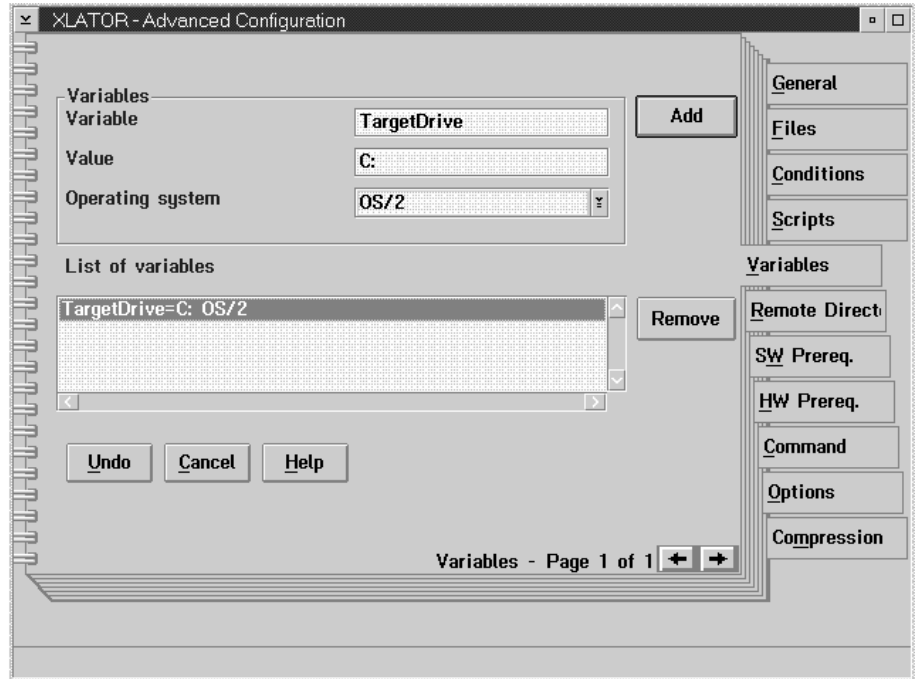


Figure 109. XLATOR - advanced configuration notebook: variables page

In this case, the only variable is the target drive on which XLATOR is to be installed.

To change the target drive:

- Select **TargetDrive=E** in the List of variables.
- Change **E** to **C** in the **Value** entry field.
- Select **OS/2** in the Operating system list.
- Click on the **Add** push button.

The **TargetDrive=C** item is displayed in the List of variables list.

- 10 Close the XLATOR - Advanced Configuration Notebook.

Cataloging the Software Object

After confirming that XLATOR contains the correct files and changing the target drive, you are ready to catalog XLATOR.

1. Select XLATOR in the Software Object Profiles window.
2. Select **Catalog** from **Catalog** menu bar choice, as shown in Figure 110 on page 192.

Replicating an Installation with DiskCamera

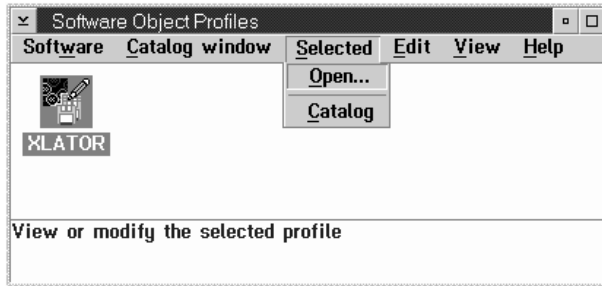


Figure 110. Software object profile window

3. A progress indicator is displayed, as shown in Figure 111.

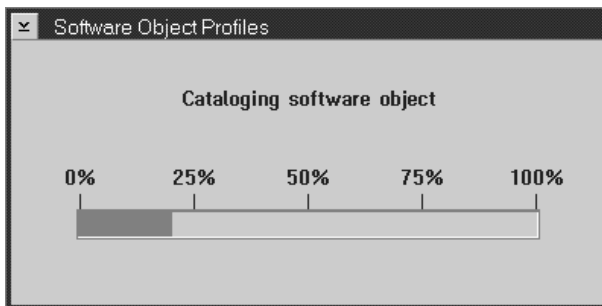


Figure 111. Software object profiles progress indicator

4. An informational message is displayed, as shown in Figure 112.



Figure 112. Software object profiles informational message

5. Click on the **OK** push button.

An object for XLATOR is displayed in the TME 10 Software Distribution for OS/2 Catalog window, as shown in Figure 113 on page 193.

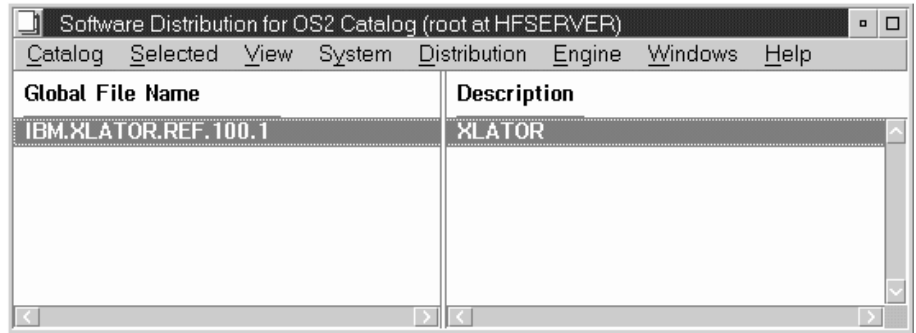


Figure 113. TME 10 Software Distribution for OS/2 catalog window

For information about how to authorize clients to install cataloged applications (such as XLATOR) on request, see “Scenario 6: Defining and Authorizing a Client to Install Software Objects” on page 205.

For information about how to request installation of a cataloged application (such as XLATOR) from an authorized client, see “Scenario 7: Installing an Application at a Client (Pull Installation)” on page 211.

Note that the preparation site (MYPREP) and the target workstation (HFCLIENT) should have the same fix level of the operating system installed.

Scenario 5: Installing an Application from the Internet using DiskCamera

In this scenario, you will:

- 1 Download an application from the Internet
- 2 Run the AntiVirus check
- 3 Take a “before” picture of the drive where the application is to be installed
- 4 Install the application, using DiskCamera to record the details of the installation
- 5 Take an “after” picture of the drive
- 6 Build the software object and catalog the application.

Downloading the Application from the Internet

The application used in this example is the PM Graphical File Management utility, which is also called FM/2. A sample window from FM/2 is shown in Figure 114.

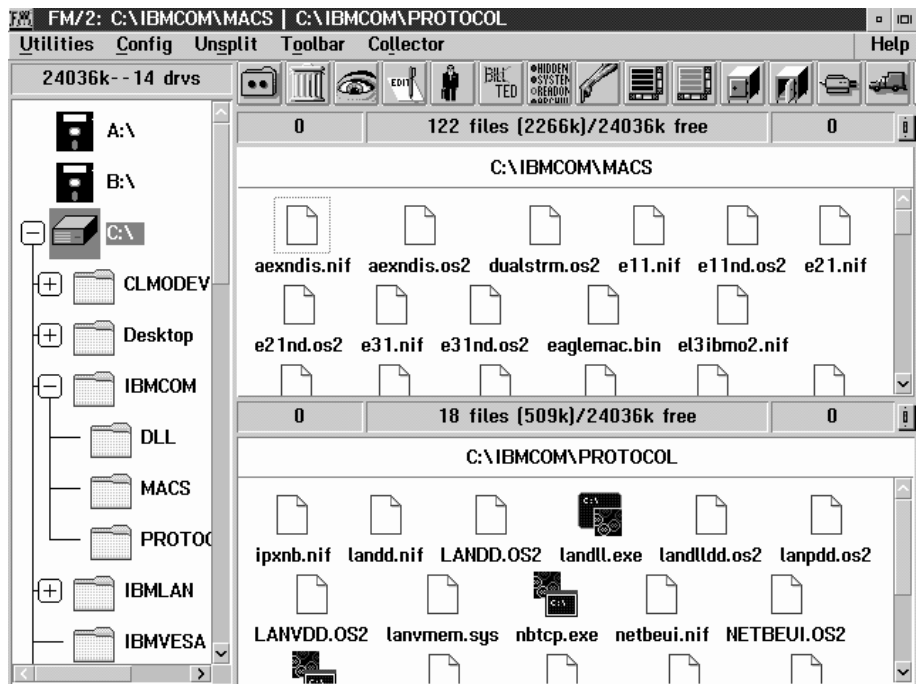


Figure 114. Sample File Manager window

To download the utility, follow these steps, using your Web Explorer:

- 1 Open the Software Library - OS/2 Software page at the URL:

<ftp://hobbes.nmsu.edu/os2/diskutil>

as shown in Figure 115.

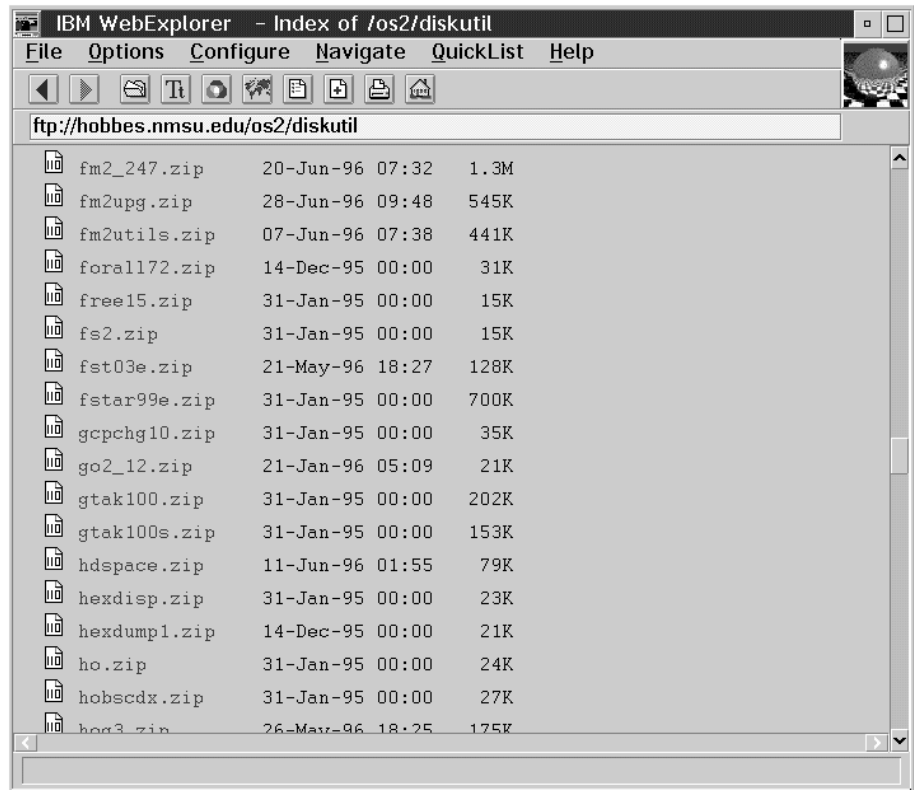


Figure 115. Downloading the application

Select the fm2_247.zip utility and download it to a temporary directory. Be sure you download it on a drive *different* from the drive where you will install the application using DiskCamera. In this example, the application is downloaded to F:\TMP.

- 2 Unzip the program with any shareware or commercial unzip program. If you do not have one, you can download it from the Info-Zip World Wide Web page at the URL:

<http://quest.jpl.nasa.gov/Info-ZIP/UnZip.html>

selecting one of the sites containing programs in OS/2 ready-to-run binary format. For example, you can choose:

www.hensa.ac.uk (UK)

which points to the URL:

<http://www.hensa.ac.uk/ftp/mirrors/uunet/pub/archiving/zip/0S2/>

from which you can download:

unz512x2.exe

Run the unz512.exe self-extracting package to get the unzip.exe file that you need to unpack the fm.zip and install it. Place the unzip.exe under a directory specified by the PATH statement in your CONFIG.SYS. Unpack the fm2_247.zip into the F:\TMP directory with the command:

```
unzip fm2_247.zip
```

You now have the FM/2 files in the TMP directory.

Now you are ready to run IBM AntiVirus.

Running IBM AntiVirus

To run a virus check on the program just downloaded, perform the following steps:

1. Start the IBM AntiVirus program. The AntiVirus window is displayed, as shown in Figure 116.

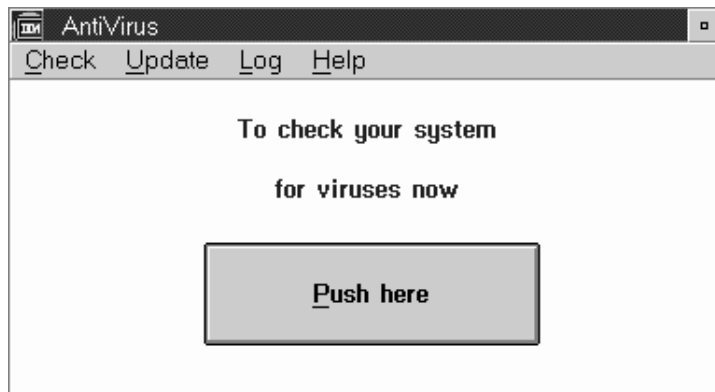


Figure 116. AntiVirus window

2. Select **Check system...** in the **Check** menu choice.

The Check system for viruses window is displayed, as shown in Figure 117 on page 197.

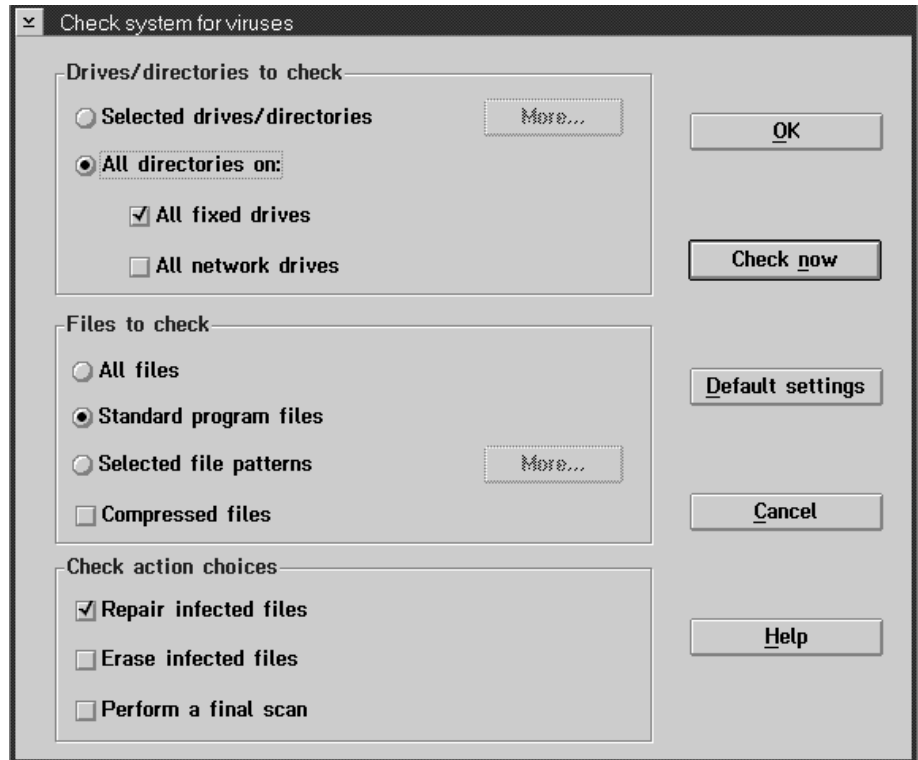


Figure 117. Check system for viruses window

3. Check the **Selected drives/directories** radio button and click on the first **More...** push button in the same group box. The Select drives/directories window is displayed, as shown in Figure 118 on page 198.

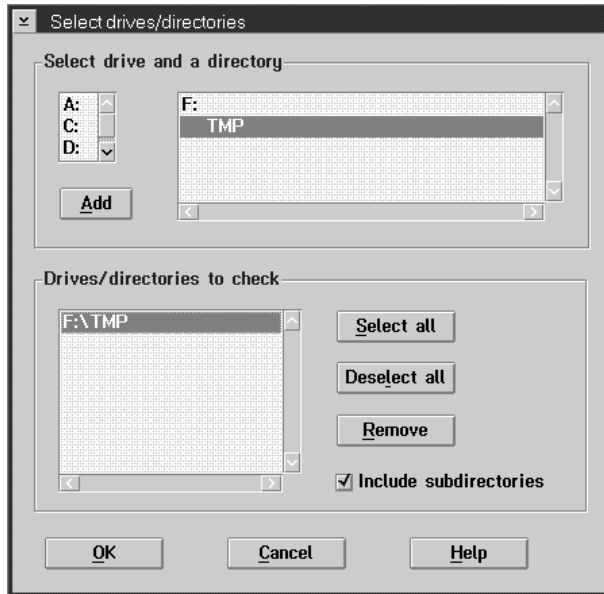


Figure 118. Select drives/directories window

4. Set the drive and directory to be checked, in this example F:\TMP, and click on the **OK** push button. The Check system for viruses window is displayed again, as shown in Figure 117 on page 197.
5. Click on the **OK** push button. The AntiVirus window is displayed again, as shown on Figure 116 on page 196.
6. Click on the **Push here** push button. The AntiVirus program starts. A message is displayed indicating that the system scanning is in progress, then an informational message follows showing that the scan is completed.
7. Click on the **OK** push button and close the AntiVirus program.

Now you are ready to take the “before” picture and install the product.

Taking a “Before” Picture

Set aside a dedicated drive (in this example, E) for DiskCamera use only, to be sure that the only changes DiskCamera records are those resulting from the installation. For performance reasons, it is important that the drive used for DiskCamera contain as few files as possible.

To take the “before” picture, open the Create a software object profile using DiskCamera window by following the first three steps of “Taking a “Before” Picture” on page 181.

This window is displayed in Figure 119 on page 199.

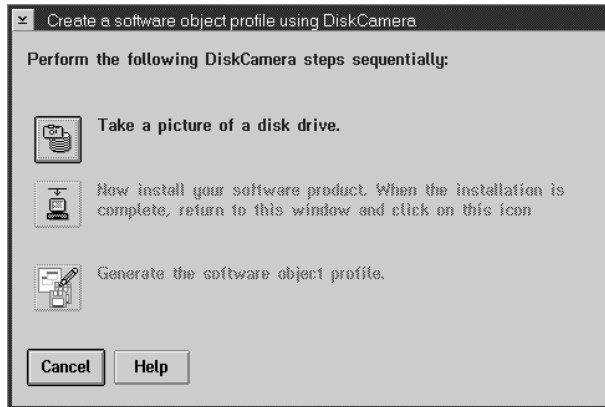


Figure 119. Create a software object profile using DiskCamera window

- 1 Click on the first button (it is the only one available).

The Take a picture of a disk drive window is displayed, as shown in Figure 120.



Figure 120. Take a picture of a disk drive window

- 2 Select the **E** drive and click on **OK** push button. The first picture starts.

A full-screen window is displayed showing which files are being monitored, this operation could take several minutes.

An informational message informs you when the disk picture has been successfully completed.

- 3 Click on **OK**. The Create a software object profile using DiskCamera window is displayed again.

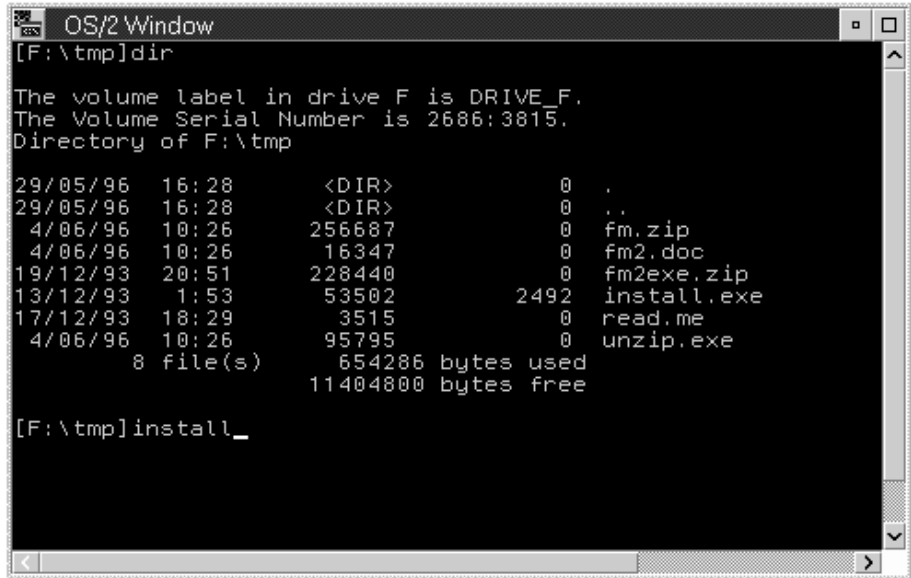
Note that the first button of this window is now checked with a red check mark to indicate that the first step of the DiskCamera process is completed.

Now you can install the application, downloaded from Internet, at the preparation site.

Installing the Application on the Disk Camera Drive

Run the installation program to install the application on the DiskCamera drive. Follow these steps:

- 1 Open an OS/2 window, go to the F:\TMP directory, and run the install program `install.exe`, as shown in Figure 121.



```
OS/2 Window
[F:\tmp]dir

The volume label in drive F is DRIVE_F.
The Volume Serial Number is 2686:3815.
Directory of F:\tmp

29/05/96  16:28      <DIR>          0      .
29/05/96  16:28      <DIR>          0      ..
 4/06/96  10:26    256687          0      fm.zip
 4/06/96  10:26    16347          0      fm2.doc
19/12/93  20:51    228440          0      fm2exe.zip
13/12/93   1:53    53502        2492      install.exe
17/12/93  18:29     3515          0      read.me
 4/06/96  10:26    95795          0      unzip.exe
      8 file(s)        654286 bytes used
                        11404800 bytes free

[F:\tmp]install_
```

Figure 121. Issuing the install command

- 2 The FM/2 Installation window is displayed, as shown in Figure 122 on page 201.

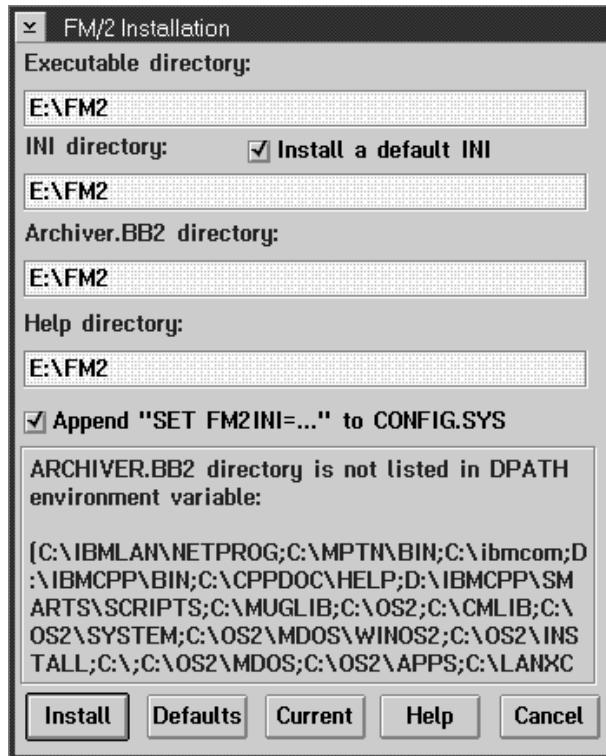


Figure 122. FM/2 installation window

- 3 Enter the name of your target directory E:\FM2.
- 4 Answer **Yes** to the message that asks if you want to create the FM2 directory.
- 5 Select the two check boxes to add a default INI file and to add to CONFIG.SYS the setting of the FM2INI environment variable. Click on **Install**.
- 6 The installation program prompts you to decide whether or not you want a WPS object on your desktop. Select **Yes** and close the FM/2 - Setting notebook that is displayed.
- 7 Verify that the icon of FM/2 is now on your desktop.
- 8 Check your CONFIG.SYS. Notice that a line with the statement SET FM2INI=E:\FM2 was added at the end of the file.
- 9 To complete the installation, update the PATH, DPATH, and HELP statements, adding the E:\FM2; directory.
- 10 Save and close your CONFIG.SYS.

Taking an “After” Picture

After the installation is completed, the next step is to take the “after” picture of the installation drive and the files being monitored. Note that it is important to be sure the installation is complete before taking the “after” picture; if you need to change the installation, you will have to restart the DiskCamera process from the beginning.

It is also important to be sure all newly created and changed files are as you want them, and that no other processes running on your workstation have made unwanted changes.

To take the “after” picture of the E drive and of the system files, follow the following steps:

- 1** In the Create a software object profile using DiskCamera window (shown in Figure 99 on page 183) click on the second button to notify DiskCamera that the installation has been completed. A confirmation message is displayed. Click on **OK** to continue.

Note that the second button of this window is now checked with a red check mark to indicate that the second step of the DiskCamera process has been completed.
- 2** In the Create a software object profile using DiskCamera window (shown in Figure 99 on page 183) click on the third button (it is the only one available) to generate the software object profile.
- 3** The Software Object Profile Generator window is displayed.
Fill in the fields of this window, as shown in Figure 123 on page 203.

Software Object Profile Generator

Software Object Profile

Name

Object label in catalog

Component name

Level

Version

☒ Insert object in catalog, from where it can be distributed

Enter the software object profile Version for the catalog label

Figure 123. Software object profile generator window

Name	The name to appear on the object's icon in the Software Object Profile and Catalog windows. This is also the first part of the software object's name in the software distribution catalog.
Component name	The name of the component of the product.
Level	A numeric field, normally the maintenance level of the product. It will be the sixth part of the software object's name in the software distribution catalog.
Version	A numeric field, normally the version of the product. It will be the seventh part of the software object's name in the software distribution catalog.

Leave checked the Insert object in catalog check mark to catalog the software object.

4 Click on **OK**.

DiskCamera takes the “after” picture of the E drive and of the system files. It compares the “before” and “after” pictures and uses the differences to build and catalog the software object for FM2.

A full-screen window is displayed showing which files have changed. This operation could take several minutes.

- 5 A progress indicator is displayed, as shown in Figure 82 on page 171.
- 6 Then an informational message is displayed, as shown in Figure 124.

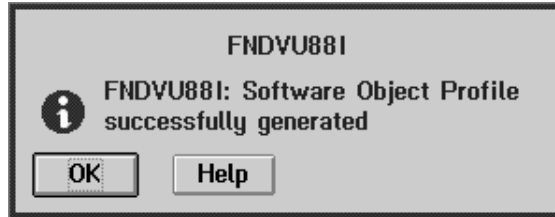


Figure 124. Successful profile creation message

- 7 Click on the **OK** push button. The Create a software object profile using DiskCamera window is displayed, as shown in Figure 125.

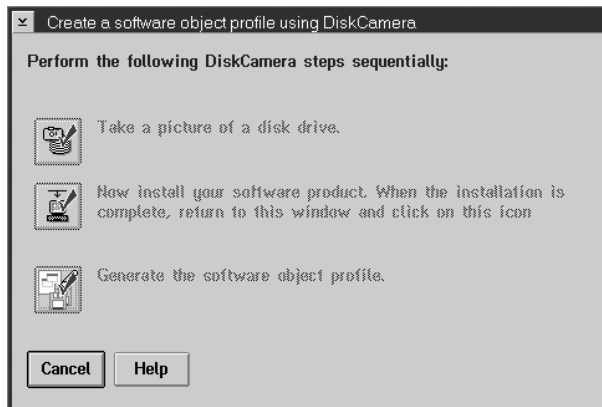


Figure 125. Create a software object profile using DiskCamera

In this window all the buttons are checked. This means that the DiskCamera process is completed.

- 8 Click on **Cancel**.

The creation of the FM/2 software object profile has been completed.

The icon for FM2 is displayed in the Software Object Profiles window, as shown in Figure 126 on page 205.

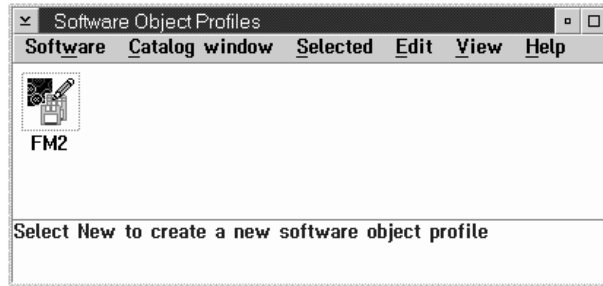


Figure 126. Software object profiles window with FM2 added

At this point the FM2 software object is cataloged and ready to be installed.

Scenario 6: Defining and Authorizing a Client to Install Software Objects

In this scenario you first define a client workstation and then you authorize it to install software objects. Then from this workstation, you install a software object, using pull installation.

By default, a software object created at a server can be installed on request, without authorization, by other servers, but not by clients. A software object created at a client can be installed on request, without authorization, by that client and by servers, but not by other clients. You can change that default, and make all software objects available to be installed on request from all clients.

In this scenario the default has not been changed. You define the **HFCLIENT** client workstation, as a target, from your **HFSERVER** server workstation and authorize the client to install a software object from the catalog of **HFSERVER**.

Authorizing Client Installation

To define a target workstation, follow these steps:

- 1 Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon View folder is displayed, as shown in Figure 127 on page 206.

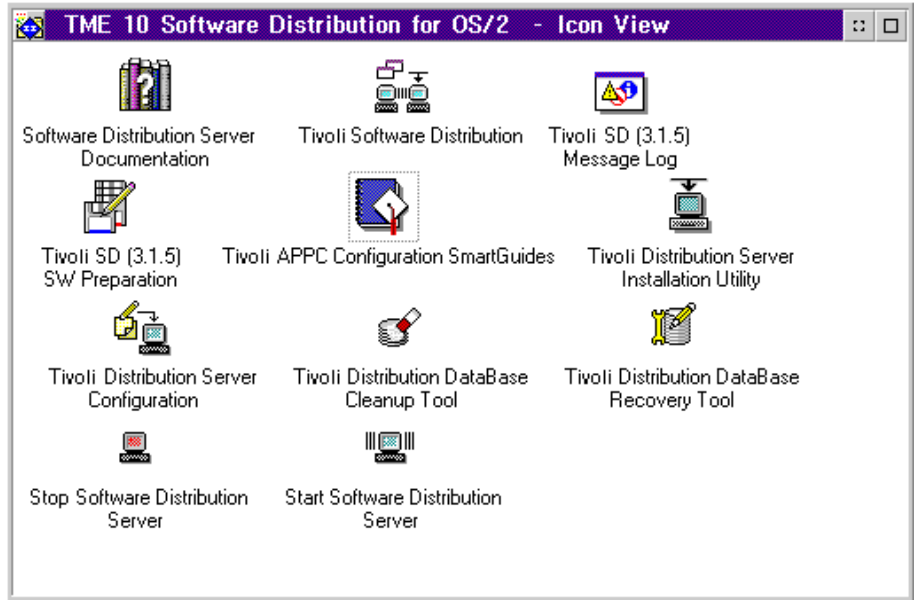


Figure 127. TME 10 Software Distribution for OS/2 - icon view folder

- 2 Double-click on the **TME 10 Software Distribution** icon.

The TME 10 Software Distribution for OS/2 Logon window is displayed, as shown in Figure 128 on page 207.



Figure 128. TME 10 Software Distribution for OS/2 logon window

- 3 Enter your user ID and password, and click on the **Logon** push button.
The TME 10 Software Distribution for OS/2 Catalog window is displayed.
- 4 In the TME 10 Software Distribution for OS/2 Catalog window, select **Targets** from the **Windows** menu bar. The TME 10 Software Distribution for OS/2 Targets window is displayed, as shown in Figure 129.

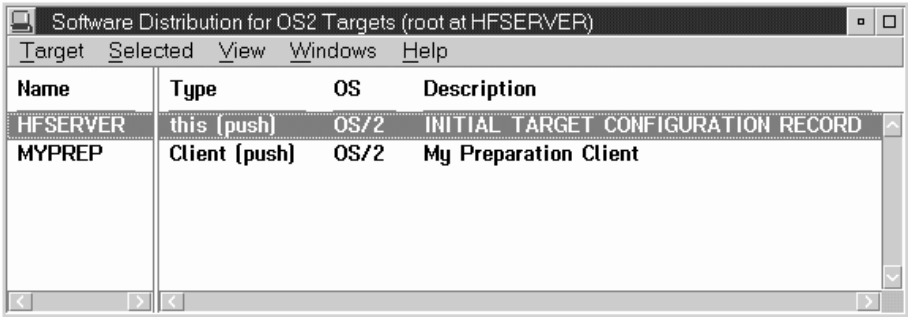


Figure 129. TME 10 Software Distribution for OS/2 targets window

Authorizing Client Installation

- 5 Select **New target...** from the Target menu bar choice. The New Local Target window is displayed, as shown in Figure 130 on page 208.

The screenshot shows the 'New Local Target (root)' dialog box. It has a title bar with a minimize button and the text 'New Local Target (root)'. The dialog contains several input fields and a 'Change management' section. The 'Name' field is 'HFCLIENT' and the 'Description' is 'My new client'. Under 'Change management', the radio button 'Initiated from focal point or any target (push)' is selected. The 'Target address' is 'HFCLIENT' and the 'Lan address' is '400010390000'. The 'Target type' is 'Client', 'Server name' is 'HFSERVER', 'Target OS' is 'OS/2', and 'Domain Address' is 'SERVER'. The 'Access Key' is 'NONE'. There are empty fields for 'Password' and 'Verify Password'. At the bottom, there are buttons for 'Details...', 'Periods of activity...', 'Parameters...', 'Hardware...', 'Log...', 'Protocol Type...', 'OK', 'Cancel', and 'Help'.

Figure 130. New local target window

- 6 Fill in the fields as shown in Figure 130.

Note that the target address and LAN address fields must be filled in with specific data related to the client workstation that you are defining.

Name	Name of the target workstation.
Description	Brief description of the target workstation.
Target address	Address of the target workstation (depends on the communications protocol that you choose by clicking on the Protocol Type... push button).
LAN address	LAN address of the target workstation.
Target type	Type of the target workstation.
Server name	Name of the distribution server for the target workstation.
Target OS	Operating system of the target workstation.

- Domain Address** Address of the domain to which the target belongs.
- Access Key** Access key assigned to the target.
- Password** Password to be used to access the target. You must enter it every time you issue a command.
- Verify Password** A check of the password entered under **Password**.

- 7 Click on the **OK** push button and the TME 10 Software Distribution for OS/2 Targets window is displayed, as shown in Figure 131.

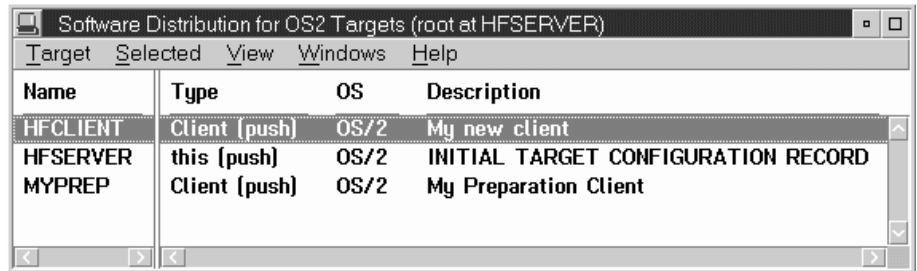


Figure 131. TME 10 Software Distribution for OS/2 targets window

The **HFCLIENT** client workstation is now defined.

Now you are ready to authorize this client to install software objects from the **HFSEVER** server workstation.

- 8 Select **Catalog** from **Windows** menu choice. The TME 10 Software Distribution for OS/2 Catalog window is displayed, as shown in Figure 132.

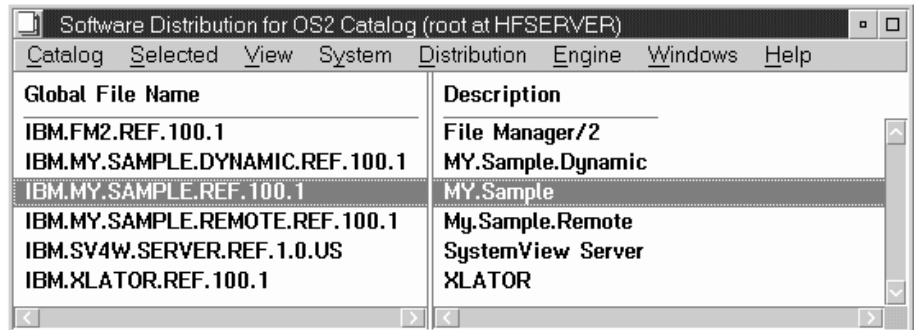


Figure 132. TME 10 Software Distribution for OS/2 Catalog window

- 9 Select the **IBM.MY.Sample.REF.100.1** software object.
- 10 Select **Authorize...** from the **Selected** pull-down. The Authorize Files window is displayed, as shown in Figure 133 on page 210

Authorizing Client Installation

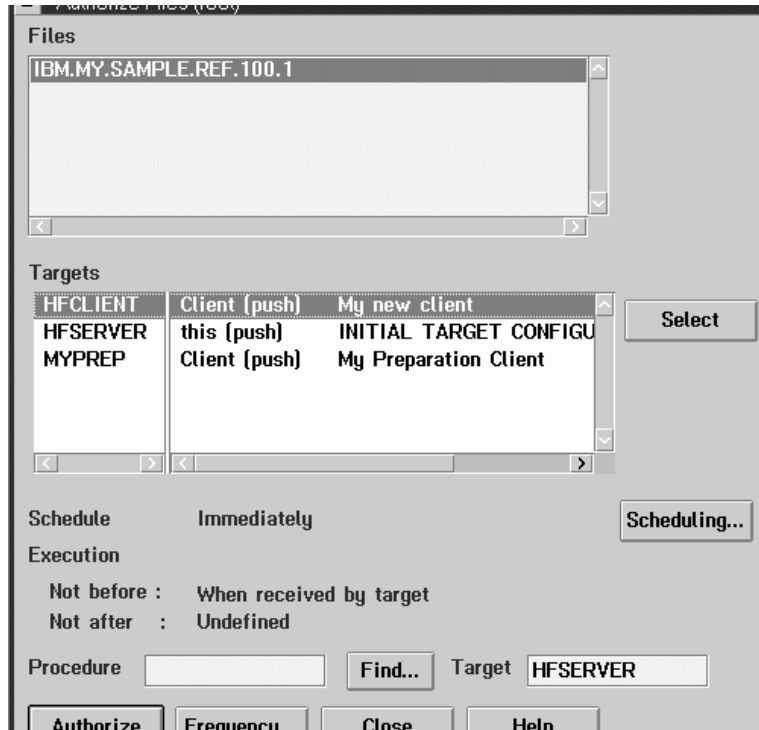


Figure 133. Authorize files window

- 11 Select **HFCLIENT** in the Targets list and click on the **Authorize** push button. The Correlators window is displayed, as shown in Figure 134.

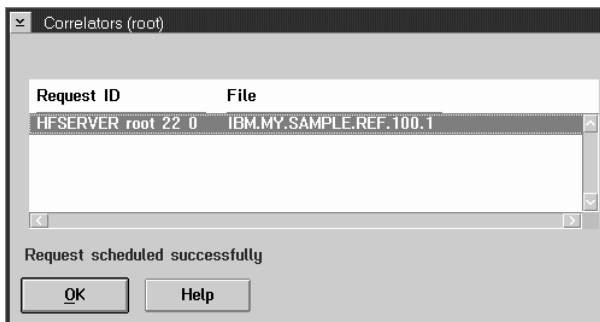


Figure 134. Correlators window

- 12 Click on the **OK** push button. The Authorize Files window is displayed again.
- 13 Click on the **Close** push button.

Now the **HFCLIENT** client workstation is authorized to install the **IBM.MY.Sample.REF.100.1** software object.

Scenario 7: Installing an Application at a Client (Pull Installation)

After a client has been authorized to install an application, a user at the client can request the installation. To request installation of an authorized application from a client, follow these steps at the client workstation:

- 1 Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon View folder is displayed, as shown in Figure 135.

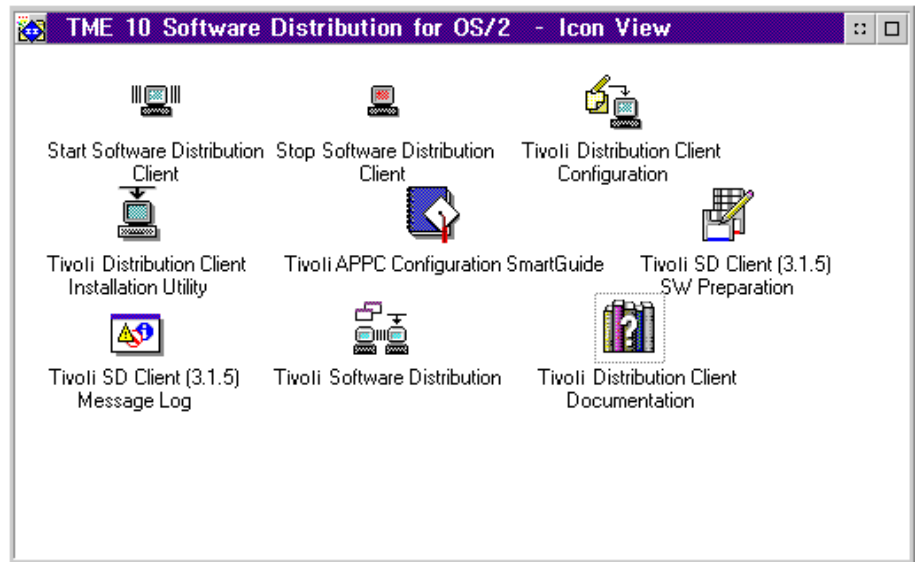


Figure 135. TME 10 Software Distribution for OS/2 - icon view folder

- 2 Double-click on the **TME 10 Software Distribution** icon.

The TME 10 Software Distribution for OS/2 Logon window is displayed, as shown in Figure 136 on page 212.

Installing at a Client (Pull)



Figure 136. TME 10 Software Distribution for OS/2 logon window

- 3 Enter your user ID and password, and click on the **Logon** push button. The TME 10 Software Distribution for OS/2 Catalog window is displayed, as shown in Figure 137.

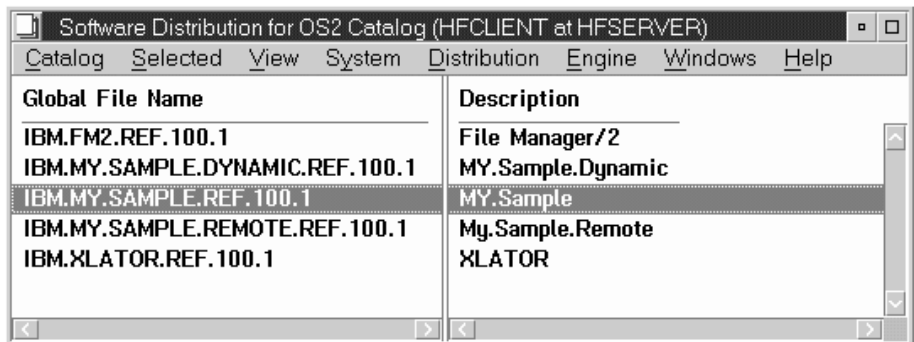


Figure 137. TME 10 Software Distribution for OS/2 Catalog window

- 4 Select the **IBM.MY.Sample.1.REF.100.1** software object.

- 5 Select **Install** from the **Selected** pull-down. The Install Change Files window is displayed, as shown in Figure 138 on page 213.

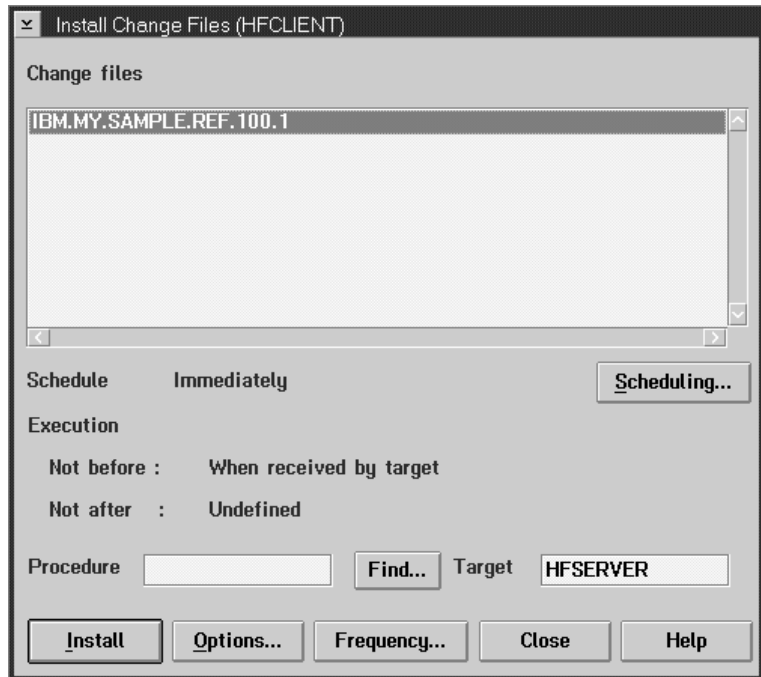


Figure 138. Install Change Files window

- 6 Click on the **Install** push button. The Correlators window is displayed, as shown in Figure 139.

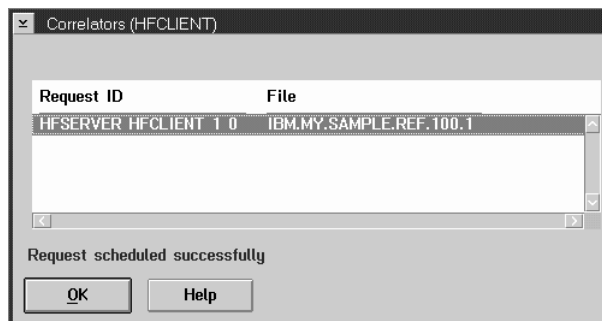


Figure 139. Correlators window

- 7 Click on the **OK** push button.

The pull installation of the **IBM.MY.Sample.1.REF.100.1** software object on the **HFCLIENT** client workstation is completed.

Chapter 17. Setting Up for CID Preparation

Using the CID (configuration, installation, and distribution) methodology, you can install CID-enabled software products on unattended workstations. The CID software preparation function assists you in creating the files that are necessary to perform the unattended installation.

The CID methodology makes use of a *response file*. A response file is a file that contains predefined answers to the questions asked by the installation program during an attended installation. TME 10 Software Distribution CID software preparation can generate the response file or can use a response file that already exists.

TME 10 Software Distribution CID software preparation is based on the information contained in a file called an *application definition file (ADF)*, which defines how to configure and install the specific product. Before you can prepare a CID-enabled software product for distribution, you must have an ADF for the software product. Various products ship ADFs on their distribution media. In some cases you might choose to create your own ADFs. See Appendix F, "Application Definition File Considerations" on page 275 for considerations about ADFs.

CID Software Distribution Environment

The CID software distribution environment includes workstations playing four different roles, as shown in Figure 140.

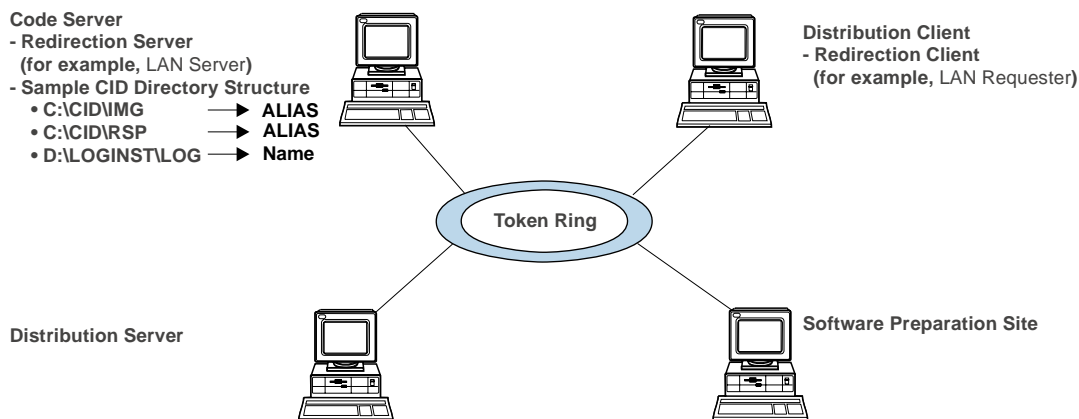


Figure 140. Logical LAN environment for CID distribution

- *Code Server*

This workstation is a repository for the code images and the response files of the products to be installed, and for a log of messages from CID installations, stored in a predefined directory structure. TME 10 Software Distribution does not necessarily have to be installed on the code server workstation. In deciding whether or not to put the code server on the same workstation as the distribution server, the factors to consider include storage requirements for the code images and whether multiple distribution servers will be sharing the code server.

On the code server you must have installed the redirection software needed to allow clients to access the images and the response files during CID installation. The redirection software can be any one of the following:

- IBMLS (LAN Server)
- NFS (Network File System) Server (distributed with TCP/IP for OS/2)
- SRVIFS (distributed with LAN Server)
- NetWare Server

- *Distribution Server*

This is a workstation where the TME 10 Software Distribution server is installed. The catalog of software available to be installed is maintained at this workstation. The administrator initiates installation of software to distribution clients from this or any other TME 10 Software Distribution server workstation.

- *Preparation Site*

This is a workstation where the TME 10 Software Distribution server or client, including the preparation site component, is installed. At the preparation site, users prepare the software to be installed and catalog it on the distribution server.

- *Distribution Client*

This workstation is the target of the CID installation. It can be unattended during a CID installation. The distribution client has the TME 10 Software Distribution Client and the redirection software needed to access the code images and response files that are on the code server. The redirection software can be any one of the following:

- LAN Requester
- Network File System (NFS) client
- SRVIFS client
- NetWare client



In the scenarios in the rest of this chapter, the code server, distribution server, and software preparation site are the same physical workstation. Figure 140 on page 215 shows a LAN environment where these functions are on separate workstations.

Overview of the Steps

Your CID setup, preparation, and installation activities will take place in the following phases in the following sequence:

- 1 Running the code server setup utility at the preparation site.** You perform this step only once. See “Running the Code Server Setup Utility at the Preparation Site.”
- 2 Completing the setup on the code server workstation** (required if the redirection software is not IBM LAN Server). You perform this step only once. See “Completing the Setup on the Code Server Workstation” on page 224.
- 3 Adding product images and response files on the code server after initial setup.** See “Uploading Software Images to the Code Server” on page 226 for automated uploading using the upload software images utility, and “Copying Products on the Code Server” on page 231 for manual copying on the code server.
- 4 Preparing and cataloging the software products to be distributed.**
- 5 Distributing and installing the software products.**

Running the Code Server Setup Utility at the Preparation Site

This section shows how to specify, at the TME 10 Software Distribution preparation site, the redirection mechanism you will use during CID installation, the name of the code server, and aliases for the CID directory (where code images and response files are stored) and the LOGINST directory (where installation messages are stored). If you do not perform this step, you are prompted to do it during CID software preparation.

If your redirection software is IBM LAN Server, in this step TME 10 Software Distribution performs automated code server setup.

To start the code server setup, perform these steps:

- 1** Double-click on the product icon.

The TME 10 Software Distribution for OS/2 - Icon View folder is displayed, as shown in Figure 141 on page 218.

Running the Code Server Setup Utility at the Preparation Site

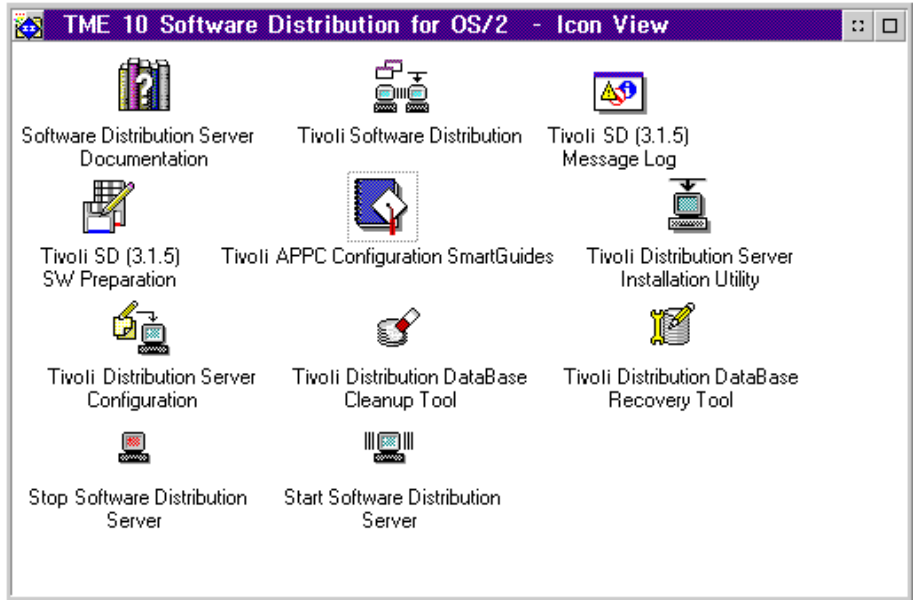


Figure 141. TME 10 Software Distribution for OS/2 - icon view folder

2 Double-click on the **Software Preparation** icon.

The Software Preparation folder is displayed, as shown in Figure 142.

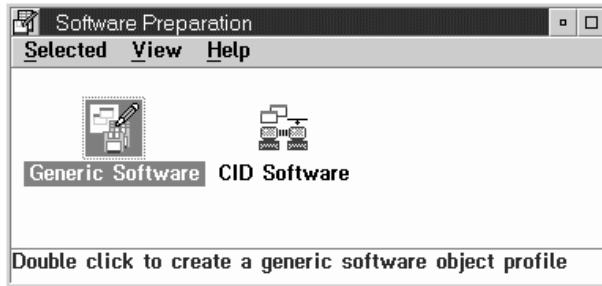


Figure 142. Software preparation folder

3 Double-click on the **CID Software** icon.

The CID Software Preparation window is displayed, as shown in Figure 143 on page 219.

Running the Code Server Setup Utility at the Preparation Site

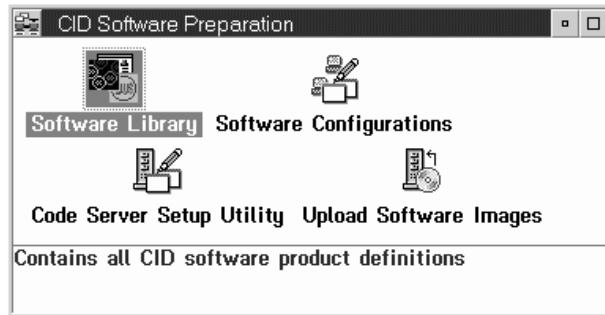


Figure 143. CID software preparation window

- 4 Double-click on the **Code Server Setup Utility** icon.

The Code Server Setup Utility notebook is displayed, as shown in Figure 144.

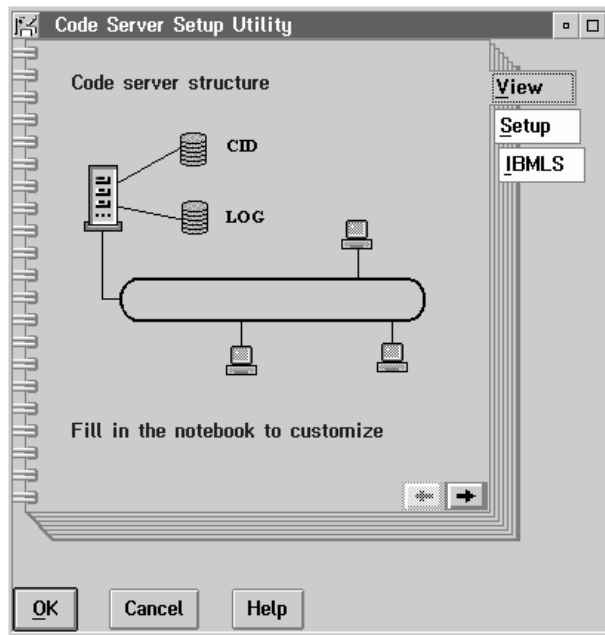


Figure 144. Code server Setup Utility Notebook

- 5 Click on the **Setup** tab or on the right arrow to proceed. Be sure you select all the listed variables and either accept or change each default value. The explanation area shows, for each selected variable, the information on how to set it.

Running the Code Server Setup Utility at the Preparation Site

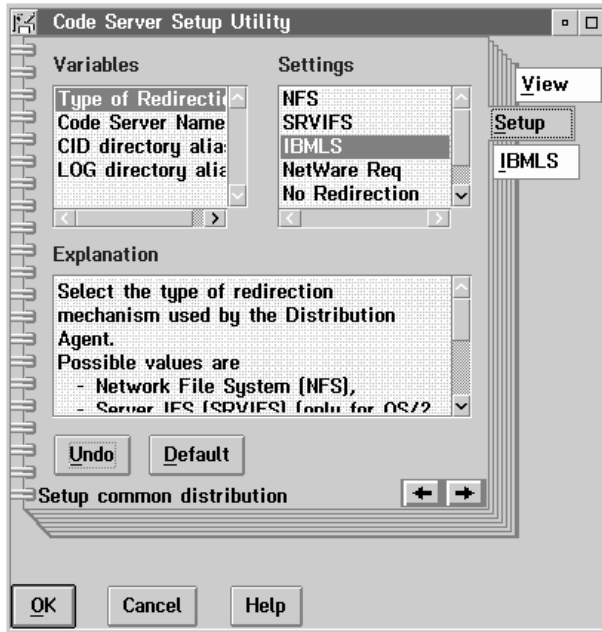


Figure 145. Type of redirection setting

Specify the type of redirection. Note that if the code images and response files are on the local workstation where the product is going to be installed, no redirection is necessary; in this case, select **No Redirection** and specify local directories for the CID and LOG aliases.

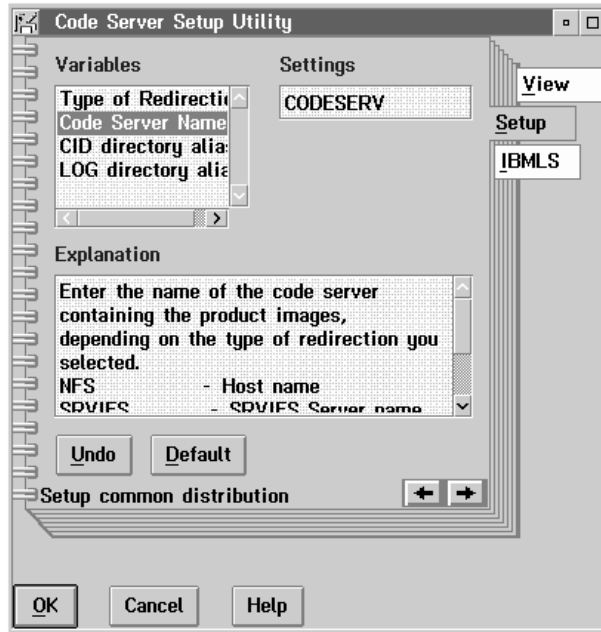


Figure 146. Code server name setting

Specify the code server name (in this example, CODESERV).

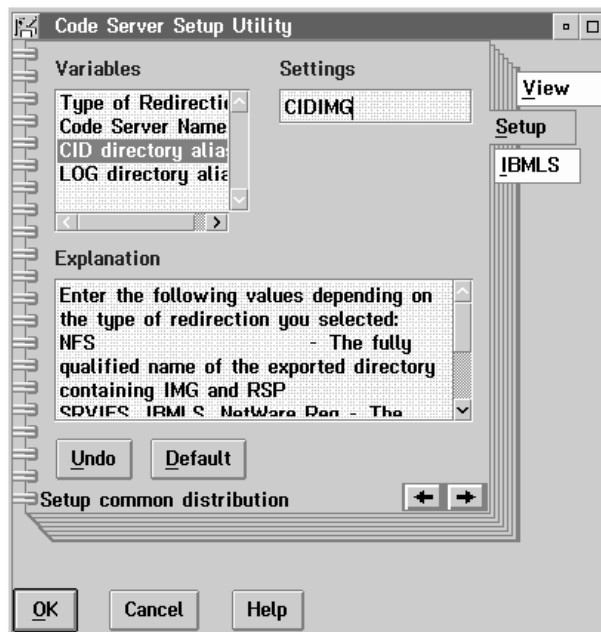


Figure 147. CID directory alias setting

Running the Code Server Setup Utility at the Preparation Site

Specify the alias for the CID directory (in this example, CIDIMG). This is the directory where code images and response files are stored.

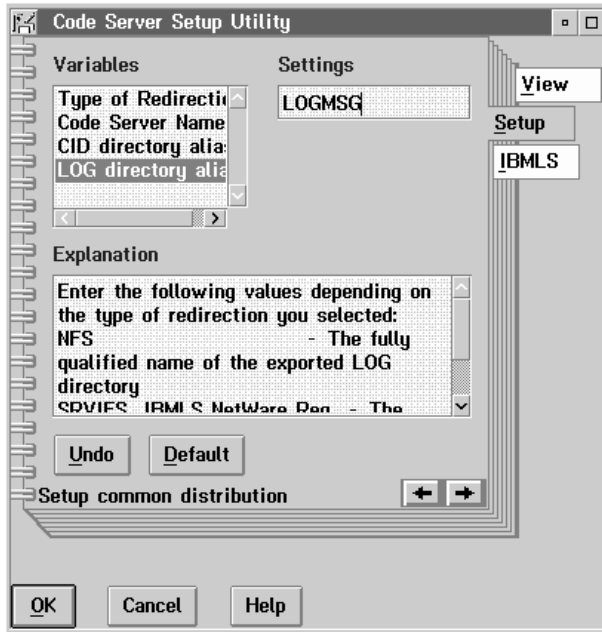


Figure 148. LOGINST directory alias setting

Specify the alias for the LOGINST directory (in this example, LOGMSG). This is the directory where messages from CID installations are to be stored.

6 Set variables for automatic code server setup (IBM LAN Server only).

If you did not select **IBMLS** as the redirection software, skip this step.

If you selected **IBMLS** as the redirection software (see Figure 145 on page 220), go to the IBMLS page, shown in Figure 149 on page 223.

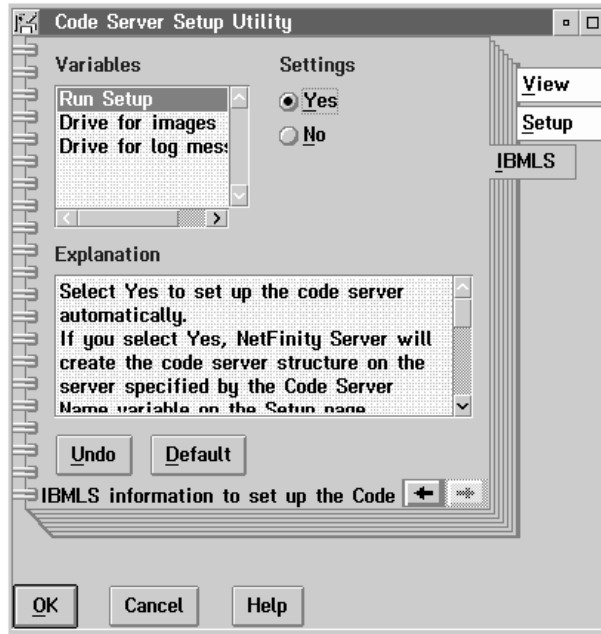


Figure 149. IBM LAN server setting

On this page you set variables to allow the code server setup utility to automatically generate and run a command file that performs the following setup activities on the code server workstation:

- Create the base CID directory structure.
- Create the aliases for the directories CID and LOGINST.
- Create access profiles for the directories.
- Authorize the IBM LAN Server user group USERS to read the files in the directory CID.
- Authorize the IBM LAN Server user group USERS to write in the directory LOGINST.

The variables are:

a Run Setup

Select **Yes** if you want to generate and immediately run the command file to set up the code server. Select **No** if you want to generate the command file and run it later. The file generated, CSSETUP.CMD, is stored in the directory:

<TME 10 Software Distribution Installation Drive>:\SOFTDIST\BIN

b Drive for images

Completing the Setup on the Code Server Workstation

The letter of the drive (on the code server workstation) on which TME 10 Software Distribution is to create the CID\IMG directory (for code images) and the CID\RSP directory (for response files).

In these scenarios use the read-only spin button to select **C** because you want TME 10 Software Distribution to create the directories:

C:\CID\IMG

C:\CID\RSP

C Drive for log messages

The letter of the drive (on the code server workstation) on which TME 10 Software Distribution is to create the LOGINST\LOG directory.

In these scenarios select **D** because you want TME 10 Software Distribution to create the directory:

D:\LOGINST\LOG

- 7** Click on **OK** to close the notebook.

If you selected **Yes** for **Run Setup** in Figure 149 on page 223 in the previous step, then automatic code server setup runs at this time.

Completing the Setup on the Code Server Workstation

If your redirection software is not IBM LAN Server, or if you chose **No redirection** when you ran the code server setup utility, perform the following steps. For LAN Server, the code server setup utility already performed these steps.

- 1** Create the CID directory structure at the code server workstation.

The code server directory structure consists of a CID directory with IMG and RSP subdirectories, and a LOGINST directory with LOG subdirectories. IMG, RSP, and LOGINST contain, respectively, the product code images, the response files, and the message log files.

Figure 150 on page 225 shows a generic view of the recommended directory structure.

Completing the Setup on the Code Server Workstation

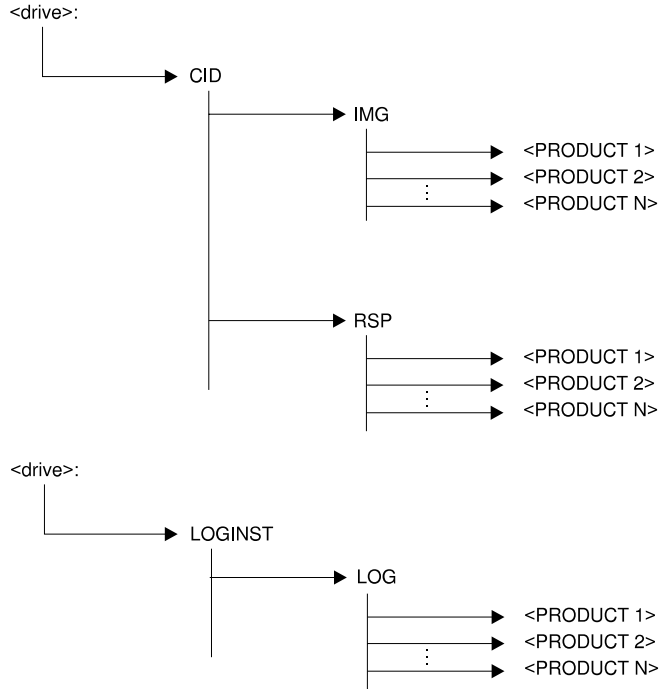


Figure 150. Recommended directory structure for CID distribution

Follow these steps to create the directory structure:

- a** Create the directories to store the code images.

Open an OS/2 window and enter the following commands from drive C:

```
MD CID
MD CID\IMG
```

- b** Create the directories to store the response files.

Enter the following commands from drive C:

```
MD CID\RSP
```

- c** Create the directories to store the log files.

Enter the following commands from drive D:

```
MD LOGINST
MD LOGINST\LOG
```

- 2** Customize the redirection mechanism by setting the name of the code server and aliases for the CID directory and LOGINST directory. (If you chose **No Redirection** when you ran the code server setup utility, skip this step.)

Customizing the redirection mechanism means setting up aliases for the directories C:\CID and D:\LOGINST, and naming the code server. The aliases

Uploading Software Images

and code server name will be used during CID software preparation (see “Running the Code Server Setup Utility at the Preparation Site” on page 217). At this stage you also set read authorization for the directory:

C:\CID

and read-write authorization for the directory:

D:\LOGINST

and define the users.

The steps to customize the redirection mechanism depend on the redirection software you installed. See the documentation for your redirection software.

Uploading Software Images to the Code Server

There are two ways to add a product to the code server. If the product is shipped with an application definition file (ADF) on CD-ROM or diskette, use the Upload Software Images utility program, which is explained in this section. Otherwise, add the product manually, as explained under “Copying Products on the Code Server” on page 231.

The Upload Software Images utility program is available to:

- Create the directory structure for the product on the code server
- Copy the product images to the code server
- Add the product to the software library
- Optionally, create a default configuration for the product and catalog the software object.

To upload the images of a product from CD-ROM to your code server, complete the following steps:

- 1 In the CID Software Preparation window, double-click on **Upload Software Images**. The Upload Software Images – Setup window is displayed, as shown in Figure 151 on page 227.

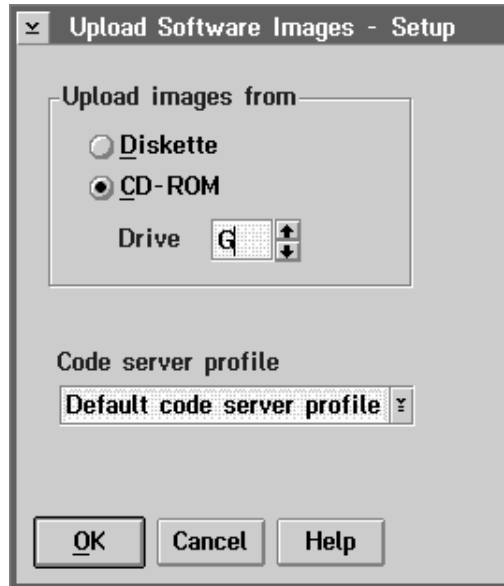


Figure 151. Upload software images – setup window

- 2 Select **Diskette** or **CD-ROM** and the letter of the diskette or CD-ROM drive. Click on **OK**. The Upload Software Images Setup – In Progress window is displayed, as shown in Figure 152.

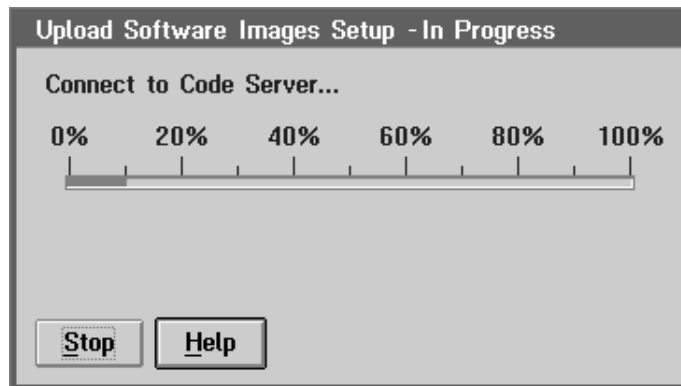


Figure 152. Upload software images setup – in progress window

- 3 The utility scans the diskette or CD-ROM for application definition files that contain all the information necessary to upload software images. The list of products is displayed in the Upload Software Images window, as shown in Figure 153 on page 228.

Uploading Software Images

Next to each product is the default name of the directory that will be created under \IMG, \RSP, and \LOG on the code server. To change the directory name, click on **Change dir...**

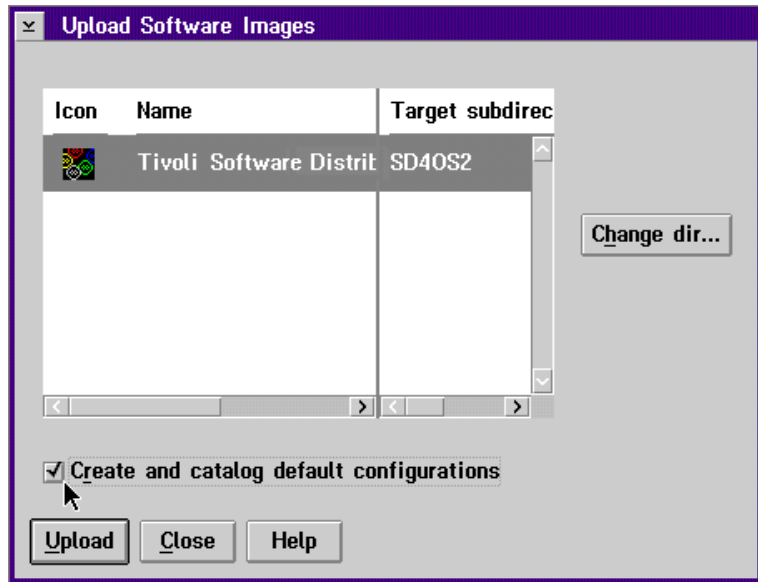


Figure 153. Upload software images window

Select the product you want to upload. In this scenario you upload, from a CD-ROM, code images of a CSD to be applied to the TME 10 Software Distribution product.

- 4 Select TME 10 Software Distribution for OS/2. Check the Create and catalog default configurations check box to create the server, client, and mobile client software object to upgrade the components. Then click on **Upload** as shown in Figure 153.
- 5 A message is displayed asking you if you want to overwrite the image directory on the server. Click on **Yes** to upgrade the code images on the code server to the CSD level, as shown in Figure 154 on page 229.

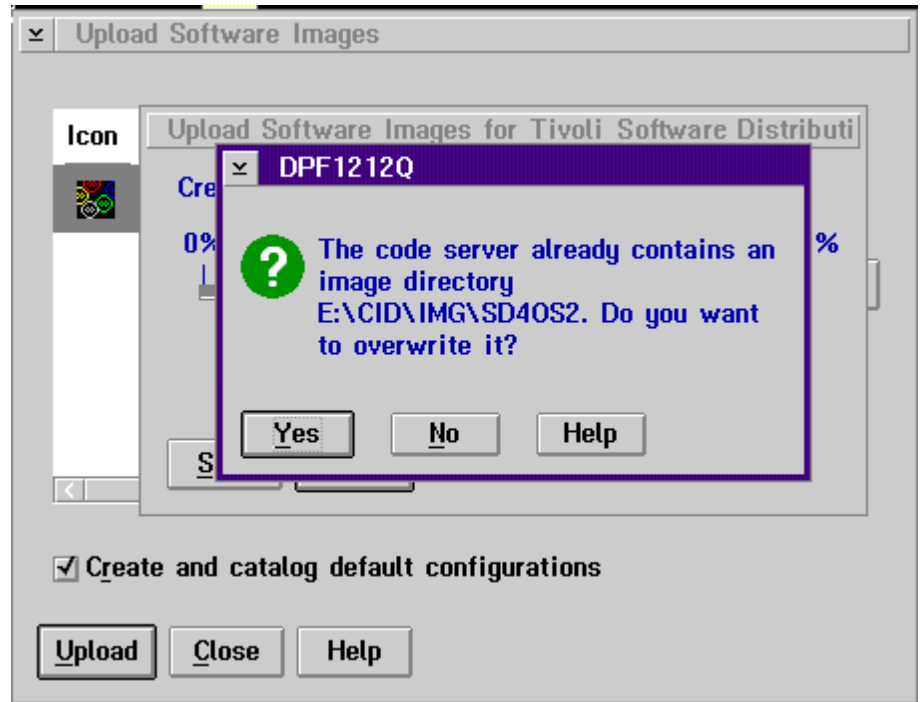


Figure 154. Upload software images overwrite message window

The Upload Software Images for TME 10 Software Distribution – In Progress Window is displayed. When the upload has completed, a message appears as shown in Figure 155 on page 230.

Uploading Software Images

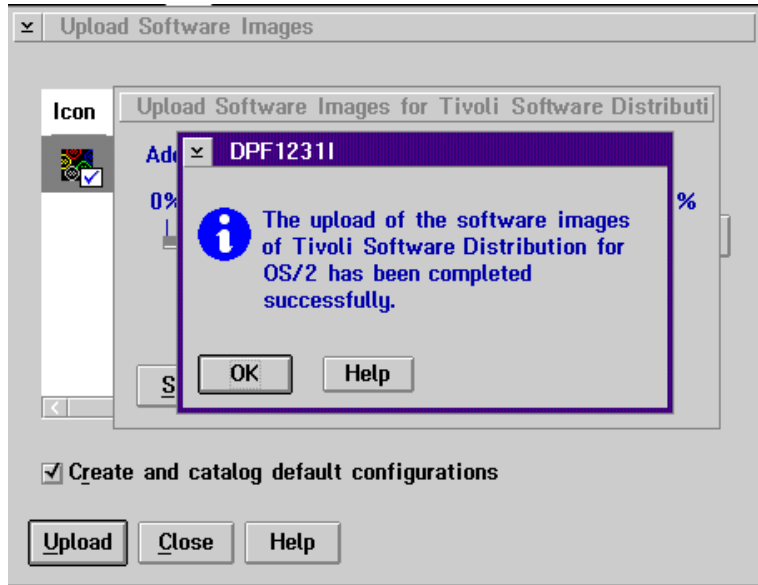


Figure 155. Upload software images message window

Click on **OK**. The Upload Software Images window is redisplayed with a modified icon for uploaded product, as shown in Figure 156.

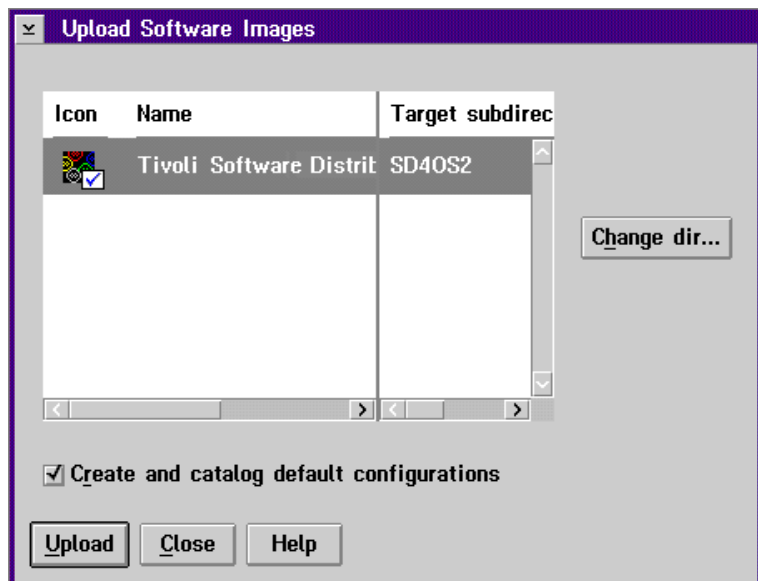


Figure 156. Upload software images message window after upload of the code images

6 Click on **Close** to finish.

- 7 In the CID Software Preparation window, double-click on **Software Library**. Note that an icon for the uploaded product has been added to the Software Library window.
- 8 In the CID Software Preparation window, double-click on **Software Configurations**.

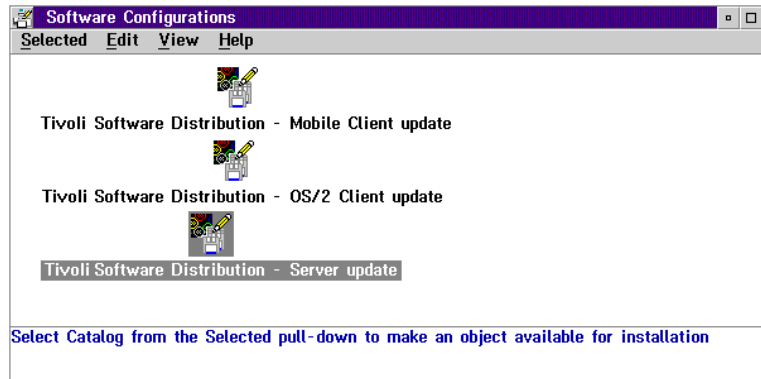


Figure 157. Software configurations window

You can see the server, client, and mobile client configurations. Three new software objects have been created in the server catalog. You can install these software objects to upgrade the TME 10 Software Distribution server, client, and mobile client workstations to the new CSD level. Then you must migrate the server and the mobile client database as described in the README file of the CSD.

Copying Products on the Code Server

If the software product you want to add to the code server is not distributed with an ADF on CD-ROM or diskette, add the product to the code server manually, as explained in this section.

This section makes use of a fictitious application program, CASHFLO. To actually run the scenario, you need to have a real application with an ADF to add to the code server.

Follow these steps:

- 1 Create the product-specific subdirectories under the base CID directory structure at the code server workstation.

Figure 150 on page 225 shows the complete CID directory structure for several software products. Under the IMG, RSP, and LOG directories already created on the code server, add a subdirectory for each software product.

Uploading Software Images

For example, assume that CASHFLO is a product with a client/server structure. To create the subdirectories to store the CASHFLO client on the code server, perform the following steps:

- a** Create the subdirectories to store the code images and response files.

Open an OS/2 window and enter the following commands from drive C:

```
MD CID\IMG\CASHFLO\CLIENT_2
MD CID\RSP\CASHFLO\CLIENT_2
```

- b** Create the subdirectory to store the log files:

Enter the following command from drive D:

```
MD LOGINST\LOG\CASHFLO\CLIENT_2
```

- 2** Copy product images and response files under the subdirectories.

Before you install any software on the clients, you must copy the product code images to the code server hard disk. The images are then made available to the clients through a redirection mechanism. Many products provide a utility for copying the images to a server. Others provide a documented procedure, often using the XCOPY command, to copy the images.

Refer to the product documentation for a detailed explanation of how to copy the product images. Copy the images for the products you intend to distribute.

For the hypothetical CASHFLO program, perform the following steps to copy the images and supplied response file from the CASHFLO CD-ROM:

- a** Insert the CD-ROM containing CASHFLO into the CD-ROM drive.

- b** Open an OS/2 window and enter the following command:

```
XCOPY <CD-ROM source drive>:\CASHFLO\CLIENT_2 C:\CID\IMG\CASHFLO\CLIENT_2 /S
```

- c** Enter the following command to make the files updatable:

```
ATTRIB C:\CID\IMG\CASHFLO\CLIENT_2\*.* -R
```

Now you are ready to add CASHFLO to the software library.

Adding Products to the Software Library

If you added a product to the code server manually, you must also add it to the TME 10 Software Distribution software library at the preparation site before you can prepare it for CID installation.

To add a software product to the library, perform these steps at the preparation site:

- 1** Start TME 10 Software Distribution from the OS/2 desktop.
- 2** Double-click on the **CID Software Preparation** icon.
- 3** Double-click on the **Software Library** icon.

The Software Library window is displayed.

- 4 Select **Software** from the menu bar.
- 5 Select **New...** from the pull-down menu if the ADF is not in the SWLIB directory.⁸
The New Software window is displayed. Enter the name of the new software product and the fully qualified name of the application definition file as shown in Figure 158.

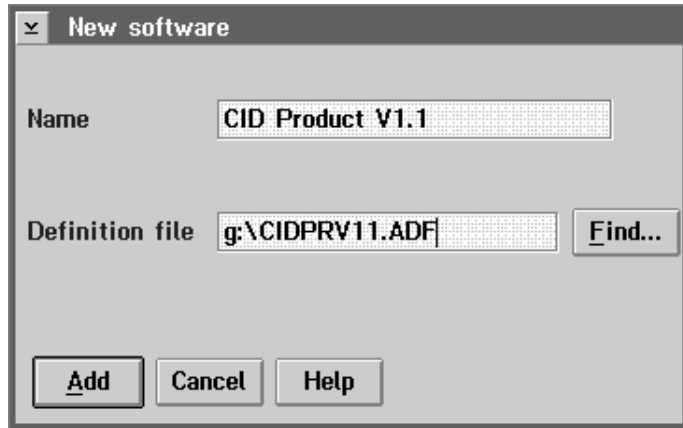


Figure 158. New software window

To search for the application definition file, in this example CIDPRV11.ADF, select **Find**.

- 6 Select **Add** to associate the application definition file with the software product being added to the TME 10 Software Distribution software library.
- 7 Select **Close** to exit.

In this example the software product CID Product V1.1 is added to the Software Library window and the related application definition file, CIDPRV11.ADF, is associated with it.

⁸ If the ADF is in the SWLIB directory, click on **Load all**, and the product is added to the software library.

Part 4. Appendixes

This part contains information to which you may need to refer from time to time while using TME 10 Software Distribution. The appendixes are:

- Appendix A, "Installing TME 10 Software Distribution by Response File" on page 237
- Appendix B, "Implementing Inventory Discovery" on page 247
- Appendix C, "Replacing the Quiesce Check" on page 253
- Appendix D, "Writing Change Control Scripts" on page 255
- Appendix E, "Writing User Exits" on page 259
- Appendix F, "Application Definition File Considerations" on page 275.
- Appendix G, "Setting Environment Variables" on page 281

Appendix A. Installing TME 10 Software Distribution by Response File

Because TME 10 Software Distribution conforms to the rules of the IBM configuration, installation, and distribution (CID) methodology, you can install TME 10 Software Distribution itself in an unattended fashion. This means that you can enter into a response file all the information for which you would otherwise be prompted during installation and configuration.

You can use response files to install TME 10 Software Distribution from the CD-ROM or from a redirected drive.

The following sections describe how to install the TME 10 Software Distribution server automatically and how to write response files that configure the TME 10 Software Distribution server options.

Using the Installation Command

The `INSTALL` command is available to install the server.

Enter `INSTALL` followed by the installation parameters at the command prompt of the CD drive. To install TME 10 Software Distribution in unattended mode, add the `/X` parameter.

Parameters can be in any order. Values can be in upper- or lowercase. You can use an equal sign (=) or colon (:) with the parameters. (The following syntax uses the : format.)

The parameters are:

- `/A:<action>`
- `/C:<catalog file name>`
- `/L1:<message log>`
- `/L2:<history log>`
- `/L3:<unattended configuration error log>`
- `/O:<originating system>`
- `/P:<product name>`
- `/R:<response file>`
- `/S:<source location>`
- `/T:<installation target location>`
- `/TU:<update CONFIG.SYS>`
- `/X <interactive flag>`

Using the Installation Command

The definition for each command prompt parameter is:

/A:<action>

Specifies the action for INSTALL to execute. Possible values for this parameter are:

D To Delete

I To Install

R To Restore

U To Update

If you omit this parameter, the installation starts interactively with all windows displayed.

If you specify D, be sure no TME 10 Software Distribution processes are active, or the deletion will fail.

/C:<catalog file name>

Specifies the name and location of the catalog file that you want to be opened automatically. This is not a required parameter; if you do not enter it, the default catalog file is opened.

/L1:<message log>

Specifies the drive, path, and file name of the installation log file for information, warning, and error messages. All lines logged to this file are prefixed with a time stamp.

/L2:<history log>

Specifies the drive, path, and file name of the installation history log file. All lines logged to the history file are prefixed with a time stamp.

/L3:<unattended configuration error log>

Specifies the drive, path, and file name of the log for error and warning messages issued during an unattended configuration.

/O:<originating system>

Specifies the originating system of the installation. The value for this parameter is *DRIVE*.

/P:<product name>

Provides the name of the product for the specified action. The value must be TME 10 Software Distribution for OS/2, enclosed in double quotation marks.

/R:<response file>

Specifies the drive, path, and name of the response file you are using to drive the installation.

/S:<source location>

Specifies the drive and path containing the source files to install. You can omit this parameter if you run INSTALL from that same location. Omit this parameter if /A is set to D for delete.

/T:<installation target location>

Specifies the drive and path where TME 10 Software Distribution is to be installed, or from which it is to be deleted if /A is set to D for delete. The default is <drive>\SOFTDIST.

/TU:<update CONFIG.SYS>

Specifies the drive and path of the CONFIG.SYS to be updated by INSTALL. Be sure to include a backslash at the end of the path; for example, code:

```
/TU:C:\
```

/X

Specifies that the action is unattended. This parameter is required for unattended installation.

Response File Layout

A response file is a flat ASCII file that consists of a series of lines separated by new line sequences. You can write or edit a response file with any editor.

Each line in a response file has a maximum length of 255 bytes.

Response file lines have the following syntax:

```
keyword = value
```

Keywords cannot contain embedded spaces.

“Response File Keywords” explains what the keywords mean. If the value of a keyword is shown in the form \$(value), substitute your own value or, for optional keywords, accept the default shown under “Response File Keywords.”

To include a comment line, place a semicolon (;) as the first character on the line.

Response File Keywords

In this section, the response file keywords are grouped into the following types:

- Installation definition
- General parameters configuration
- Software distribution configuration
- Enterprise connectivity

Installation Definition Keywords

CFGUpdate

Specifies whether the system CONFIG.SYS file is to be automatically updated at installation. The value can be:

Auto

To update

Response File Keywords

Manual

Not to update

Type: Required

COMP

Specifies the names of the TME 10 Software Distribution server or client components that you want to install. Use one COMP keyword for each component.

Possible component names for a server are:

Distribution Server

Is the TME 10 Software Distribution product.

Distribution Server GUI

Is the interactive server interface.

Preparation Site Server GUI

Adds the capability to prepare objects for software distribution by using the graphical interface.

Hw/Sw Discovery Tool

Adds the capability to do hardware and software inventory discovery on the workstation.

Distribution Server Documentation

Is the INF version of the product publications and other information files.

DeleteBackup

Specifies whether to delete any backup versions of TME 10 Software Distribution. The value can be **yes** or **no**.

Type: Required

File

Provides the new default path for the TME 10 Software Distribution file directory.

Type: Required

Overwrite

Specifies whether to automatically overwrite files during installation. The value can be **yes** or **no**.

Type: Required

SaveBackup

Specifies whether to save a backup version of TME 10 Software Distribution when it is updated. The value can be **yes** or **no**.

Type: Required

Work

Provides the new default path for the TME 10 Software Distribution data directory.

Type: Required

General Parameters Configuration Keywords

Driver

Specifies the protocols used by the workstation to communicate with other machines in the TME 10 Software Distribution network.

At least one of the following **Driver** keywords is required in the response file:

- Driver.TCPIP
- Driver.NetBIOS
- Driver.IPX
- Driver.SNA

The value is 1 if the protocol is used and 0 if it is not used. You can specify more than one driver.

Type: Required.

SystemName

Name of this system in the TME 10 Software Distribution environment.

Type: Required

Distribution Configuration Keywords

BackupArea

Is the new default path for the backup directory. The default is <drive>\SOFTDIST\BACKUP.

Type: Optional

Repository

Is the new default path for the repository directory. The default is <drive>\SOFTDIST\REPOS.

Type: Optional

ServiceArea

Is the new default path for the service directory. The default is <drive>\SOFTDIST\SERVICE.

Type: Optional

WorkArea

Is the new default path for the workarea directory. The default is <drive>\SOFTDIST\WORK.

Type: Optional

InventoryProgram

Is the new default path for the inventory program. You must specify this keyword only if you install the Hardware and Software Discovery Tool component. The default is <drive>\SOFTDIST\BIN\FNDINV.EXE.

Type: Optional

Response File Keywords

LocalDomainAddress

If you will be connecting to a TME 10 Software Distribution manager on MVS or AIX, this is the name of the domain to which this TME 10 Software Distribution server and its distribution clients belong.

Type: Optional. The default is the first eight characters of the SystemName keyword.

LocalTargetAddress

The name by which this distribution server is to be known by the MVS or AIX TME 10 Software Distribution server and the distribution clients in the LAN.

Type: Optional. The default is the first eight characters of the SystemName keyword.

Enterprise Connectivity Configuration Keywords (Server Only)

EnableRemoteCommunication

Enables communication with NetView DM for MVS or TME 10 Software Distribution for AIX.

The value can be:

Yes To enable communication with NetView DM for MVS or TME 10 Software Distribution for AIX;

No To disable such communication.

Type: Optional; the default is **No**.

You can define one or more remote server, and for each of them you must specify a set of parameters. The set of parameters are platform dependent. The keywords that belong to the definition of a server must be grouped by a progressive suffix that you put at end of the keyword.

An example of the keywords you specify in the response file to define an AIX and an MVS server follows:

```
EnableRemoteCommunication = YES
```

```
; This section defines an MVS host
```

```
DomainAddress0 = SPA001
TargetAddress0 = I9RL008A
Protocol0      = APPC
LUAlias0       = MYALIAS
ModeName0      = LU62
RemoteNetworkId0 = SPA001
RemoteLUName0  = I9RL008A
FocalPoint0    = YES
```

```
; This section defines a TME 10 Software Distribution remote server
```

```
DomainAddress1 = MYSRV
TargetAddress1 = MYSRV
Protocol1      = TCP/IP
RemoteHostName1 = myhostname
FocalPoint1    = NO
```

Note: Each server defined must be given a unique suffix on each keyword. In the example above, the MVS host has the suffix 0 and the TME 10 Software Distribution remote server has the suffix 1.

DomainAddressx

Where **x** is the sequence number of the server being defined.

If your server is to be connected to a remote software distribution manager, the name of the network to which the remote server belongs. If the remote server is a NetView DM for MVS system, this name must be equal to the network ID of the partner LU in the Personal Communications APPC configuration of this workstation. If the remote server is a TME 10 Software Distribution server, this is the domain address of the TME 10 Software Distribution server.

The name can be up to eight characters long. The characters can be letters, numbers, or the special characters @, \$, and #. An asterisk (*) implies that any of the defined domains is considered valid.

Type: Required for connection to a remote TME 10 Software Distribution server

TargetAddressx

Where **x** is the sequence number of the server being defined.

The remote server name. If the remote server is a NetView DM for MVS system, the target address is the logical unit (LU) name assigned to NetView DM for MVS; it must therefore correspond to the name of the partner LU in the Personal Communications APPC configuration of the target workstation.

If the remote server is a TME 10 Software Distribution server, this name is the target address of the TME 10 Software Distribution server.

The name can be up to eight characters long. The characters can be letters, numbers, or the special characters @, \$, and #. An asterisk (*) implies that any of the remote servers is considered valid.

Response File Keywords

Type: Required for connection to a remote TME 10 Software Distribution server

Protocolx

Where **x** is the sequence number of the server being defined.

The protocol that the remote server and the TME 10 Software Distribution server will use to communicate. Valid values are **APPC** and **TCP/IP**. **APPC** requires that Personal Communications be installed and configured on the TME 10 Software Distribution server. **TCP/IP** requires that TCP/IP be installed and configured on the TME 10 Software Distribution server.

Type: Required for connection to a remote TME 10 Software Distribution server

LUAliasx

Where **x** is the sequence number of the server being defined.

The LU alias name that identifies the software distribution program to Personal Communications when an APPC session is established with the NetView DM for MVS remote system.

The local LU alias name that identifies the target workstation, as defined in the local Personal Communications APPC configuration. Maximum length is eight characters.

The name is case sensitive. Enter it exactly as in the Personal Communications configuration.

If you do not enter a name here, Personal Communications uses the defined local LU name or local CP name in lieu of the local LU alias (if there is a working Personal Communications APPC configuration).

Type: Optional; valid for APPC connection only

ModeNamex

Where **x** is the sequence number of the server being defined.

The name of the logon mode table entry, in the MVS system, that contains the session parameters used for establishing the APPC session between NetView DM for MVS and this TME 10 Software Distribution server. Enter the name of the mode as defined in the Personal Communications APPC configuration of the target workstation. Maximum length is 8 characters.

If you do not specify a mode here or in Personal Communications, Personal Communications uses one of the default IBM-supplied modes. Note, however, that to communicate with the MVS remote system, the mode definition must contain:

```
PLU_MODE_SESSION_LIMIT=1
```

Type: Optional; valid for APPC connection only

RemoteNetworkIdx

Where **x** is the sequence number of the server being defined.

For a NetView DM for MVS remote server, the name of the network to which it belongs; for a TME 10 Software Distribution remote server, its host name. Maximum length is 8 characters.

Type: Required for connection to a remote TME 10 Software Distribution server if DomainAddress is *.

RemoteLUNamex

Where **x** is the sequence number of the server being defined.

The logical unit name of the NetView DM for MVS remote server. Maximum length is 8 characters.

Type: Required for connection to NetView DM for MVS if TargetAddress is *.

HostNameex

Where **x** is the sequence number of the server being defined.

The host name of the TME 10 Software Distribution server.

Type: Required for the connection to the remote server.

FocalPointx

Where **x** is the sequence number of the server being defined.

YES or **NO** to indicate whether the remote TME 10 Software Distribution server is a focal point. Only one server can be configured as a focal point. The default is **NO**.

Type: Optional

Appendix B. Implementing Inventory Discovery

Inventory discovery is the process that discovers what hardware and software is present on a system. You can make use of inventory information when you formulate the prerequisites for change control and distribution requests. For instance, you can specify that a software object be installed only on workstations that have at least 5MB of disk space available. Inventory information can also be used as one of the criteria for forming dynamic groups.

Inventory data is transferred from clients to the server when you do either of the following:

- From the command line interface, submit the **nvd m inv** command.
- From the graphical interface, select **Inventory** from the **Selected** pull-down menu in the Targets window.

Inventory discovery is not a mandatory part of TME 10 Software Distribution. You can configure the hardware present on a local target using the user interfaces. However, the inventory discovery process is a way of compiling this information automatically and of keeping it up to date without the administrator of your system becoming involved.

Inventory discovery can be performed on remote targets if the targets involved in the operation are connected by the server-to-server (STS) transmission protocol.

OS/2 and Windows Inventory Discovery Program

A program that takes advantage of NetFinity inventory services on workstations that run OS/2 and Windows clients is provided as an optional component at client installation. It is called **fn d inv**, and it uses NetFinity services to create output files at each target.

The discovered hardware and software data is collected and stored at the target in these files:

fn d t k inv	Stores discovered installation parameters
fn d s w inv	Stores discovered software information
fn d h w inv	Stores discovered hardware information

"Mapping to the TME 10 Software Distribution Inventory Files" on page 249 describes their formats. From these files the data is then transferred to the server's database.

Before running this inventory discovery program you must:

- 1 Make sure a NetFinity Manager workstation is installed in your network.

At the NetFinity Manager workstation:

- 2 Edit the inventory dictionary file named **DEFAULT.SID**. (You can also begin with a different dictionary file, but the final file that TME 10 Software Distribution is to use must be named **DEFAULT.SID**.) From the **Dictionary** pull-down, select **Edit** to edit the dictionary file. Select, in turn, each product you want to be detected by

Writing an Inventory Discovery Program

the TME 10 Software Distribution inventory discovery process. Edit the product to fill in two fields:

NVDM Change Object

Is the software object name that TME 10 Software Distribution is to use to identify this product.

NVDM Location Token

Is a symbol for which TME 10 Software Distribution will substitute the path in which the product is installed.

At the distribution server:

- 3** Retrieve DEFAULT.SID from the NetFinity Manager workstation.
- 4** Send DEFAULT.SID to the targets where you want to perform software and hardware inventory. Store it in <product_directory>\bin.
- 5** Identify the workstations where you want to run inventory discovery, and add the following keyword to their nvdm.cfg file:

```
INVENTORY PROGRAM:    fndinv
```
- 6** Enter:

```
nvdm inv -w target_name
```

The program gathers all hardware and software data from the target and stores it in the server's catalog.

Writing an Inventory Discovery Program

You can choose to buy an inventory discovery program from a third party or to write one yourself.

If you are writing an inventory discovery process, you need only provide the information that TME 10 Software Distribution requires. This information is presented to TME 10 Software Distribution in text files, the format of which is described in "Mapping to the TME 10 Software Distribution Inventory Files" on page 249. You do not need to provide information about all of the hardware or software present. Even a subset of the information can be used by TME 10 Software Distribution. In particular, you can choose to provide information about the hardware only or the software only.

To run the program when the inventory discovery is performed, the program name, with its complete path, must be specified in the INVENTORY PROGRAM keyword in the nvdm.cfg file. Inventory discovery is performed at each target running the program specified in the target's nvdm.cfg file.

Mapping to the TME 10 Software Distribution Inventory Files

Information about discovered software and hardware is presented to TME 10 Software Distribution at targets in text files. If you write inventory discovery yourself, you can send information directly to these files. If you choose to store the information differently, or if you are using a third-party inventory discovery procedure, you must produce a process to map your data to the inventory discovery files.

Installation Parameters Inventory File

You present information about discovered installation parameters in the installation parameters inventory file. This file is stored as <product_directory>\fndtkinv. You need root privileges to write to it.

The first time you run inventory discovery at a target, the information stored in its fndtkinv file is registered at the server. Any additional updates to the fndtkinv file are registered at the server when the inventory discovery process is run again.

The file is a text file with a fixed format. Each line defines a single installation parameter. You can define up to 1000 installation parameters. Blank lines are permitted. Comment lines begin with #.

Each parameter is presented as a keyword followed by a colon (:) or an equal sign (=). For example, the installation parameters inventory file can contain the following lines:

```
dir1: c:\mydir  
dir2= c:\mydir
```

Software Inventory File

You present information about discovered software in the software inventory file. This file is stored as <product_directory>\fndswinv. and it is created when you run inventory discovery.

A default software inventory process, called fndinv, is installed as part of TME 10 Software Distribution under <product_directory>\bin. This is the program that is executed when you run inventory discovery and creates the fndswinv file.

The file is a text file with a fixed format. Blank lines are permitted. Comment lines begin with #. You need root privileges to write to it.

Each product is described by keywords.

Figure 159 on page 250 shows a discovered software package. The package contains Version 1.3 of the messages required by a holiday booking application.

Mapping to the TME 10 Software Distribution Inventory Files

```
PRODUCT:          EUROTOURS.HOLS.MESSAGES.REF.0103
CHANGE FILE TYPE: GEN
DESCRIPTION:       Message file for Eurotours Holidays Application
PRODUCT:
  TAG:             HOLMSG
  REVISION:        1.3
  ARCHITECTURE:    OS/2
  VENDOR TAG:      IBM
  TITLE:           Message in a bottle

FILESET:
  TAG:             Base
  REVISION:        1.1.2
  TITLE:           Base feature

FILESET:
  TAG:             Message
  REVISION:        1.2
  TITLE:           Message feature
```

Figure 159. Discovered software package

Hardware Inventory File

You present information about discovered hardware in the hardware inventory file. This file is stored as <product_directory>\fndhwinv. You need root privileges to write to it.

The file is a text file with a fixed format. Each line defines a single hardware parameter. You can define up to 128 hardware parameters. Blank lines are permitted. Comment lines begin with #.

Each parameter is presented as a keyword followed by a colon (:) followed by a number. The definition of the keyword and the meaning of the numeric value depends on the construction of the software object that references the hardware prerequisites.

For example, the hardware configuration file might contain the following line to indicate that the target has a 600 MB hard disk.

```
FixedDisk.capacityMb:    600
```

TME 10 Software Distribution, Version 3.1.5 for OS/2 uses the following default keywords to identify discovered hardware devices. If you use another tool, be sure to map the hardware keywords generated to these standard keywords:

Inventory.dateAcquired	Equipment.machineModel	OperatingSystem	OperatingSystem.version
PWS.memorySizeMb	PWS.parallelPorts	PWS.serialPorts	PWS.planar/
PWS.planarFRUNumber/	PWS.busType/	PWS.biosType/	PWS.biosRevisionLevel/
PWS.biosLevelDate/	PWSType.processor/	PWSType.processorSpeed/	PWS.coProcessors
PWS.coProcessor/	PWS.language/	PWS.cacheSizeKb	Display.adapters
Display.adapterType/	Display.type/	Display.memory/	Display.colors/
Display.hozResolution/	Display.verResolution/	Display.hozSize/	Display.verSize/
Keyboards	Keyboard.adapter/	Keyboard.type/	Keyboard.countryCode/
Keyboard.subCountryCode/	Keyboard.codePage/	Printers	Printer/
Printer.driver/	Plotters	Plotter/	Mouses
Mouse.type/	Mouse.buttons/	LogicalDisk.driveLetter/	LogicalDisk.volumeLabel/
LogicalDisk.fileSystem/	LogicalDisk.availCapacityMb/	DisketteDrives	DisketteDrive/
DisketteDrive.type/	DisketteDrive.adapter/	FixedDisks	FixedDisk.capacityMb/
DASD.adapterType/	DASD.adapterAttributes/	DASD.adapterIOType/	CDROMs
CDROM/	CDROM.capacityMb/	CDROM.typeModel/	Tapes
Tape/	Tape.typeModel/	EthernetAdapters	EthernetAdapter/
EthernetAdapter.address/	TRAdapters	TRAdapter/	TRAdapter.address/
NetworkInterfaces	NetworkInterface/	IP_Interface/	IP_Network.ip_address/
IP_Network.ip_subnet_mask/	IP_Network.ip_network_name/	AdapterBoards/	MagnetoOpticalDrive/
MagnetoOpticalDrive.capacityMb/	ProtocolConverter/	Repeater/	Router/
SerialDevice/	SerialPort/	Tablet.adapter/	Tablet.type/
ParallelPort/	SCSIAdapter/	OpticalAdapter/	TTY/
TTY.speed/	PTY/	HFT/	DLC/
LVM/	LFT/	RCM/	Adapter/
Driver/	Drawer/	OtherDevice/	Tablets
TTYs	PTYs	HFTs	DCLs
LVMs	LFTs	RCMs	MultiSubchannels
ParallelPorts	SerialPorts	MultiprotocolPorts	ProtocolDrivers
SCSIAdapters	OpticalAdapters	Drawers	OtherAdapters
OtherDrivers	OtherDevices		

Defining a Filter for Hardware Keywords

You do not have to store all the hardware keywords discovered for targets in the server database. You can define a subset of those you are interested in, and when the inventory data is sent to the server they will be filtered out from among those discovered. You may also specify keywords that are not found in the default list in “Hardware Inventory File” on page 250.

To record only selected hardware keywords in the catalog, specify them in the text file called `hwfilter`, which is stored in the `<product_directory>db` directory on the server. Each line in the file can contain only one keyword which contains no blanks. The file can include comment lines that are preceded by the number sign (`#`).

Each server in a hierarchy can have a different filter file (see “Routing Inventory Information to Upstream Servers” on page 252). Downstream servers send the complete file of discovered hardware to servers above it in the hierarchy; any filtering is performed at each server the inventory file is routed to.

Running Inventory Discovery

You do not normally need to run inventory discovery more often than every time you start your target. It is probably best to include inventory discovery as part of your startup sequence.

If you add software packages without using TME 10 Software Distribution and without restarting, run the inventory discovery command so that the new software is discovered and inserted in the database.

Routing Inventory Information to Upstream Servers

Notify TME 10 Software Distribution of the discovered hardware and software by performing the inventory discovery program on your target. The inventory command sends the data from your target to the server, where it is added to the central TME 10 Software Distribution database. You must run this command to register your inventory even if your target is the server itself.

Run the inventory discovery program whenever you change the contents of the inventory files. You can run the program as part of your startup sequence or from a script or directly from the user interfaces. You need Modify Configuration authorization to issue the inventory command.

Hardware inventory information is copied to the target hardware file. Software inventory data results in the creation of a catalog entry and status record with the status of discovered, which means that the software is installed but that it was not installed by TME 10 Software Distribution. Discovered software is active and not removable.

The discovered hardware inventory is completely refreshed each time this command is run. The discovered software inventory is cumulative. That is, you can only add entries using this command. To delete a discovered software package from the inventory, you must remove it from the catalog.

The inventory command is scheduled automatically when you add a new target using either the graphical interface or the **nvdms addtg** command from the command line interface.

Routing Inventory Information to Upstream Servers

The inventory data discovered for a target can be stored and updated at all servers connected in an upstream hierarchy to the target's local server. Information will be routed and stored at all servers that have the following keyword set to YES in their base configuration file (nvdms.cfg):

```
AUTOMATIC TARGET INFO UPDATE      YES
```

The base configuration file is described in the Chapter 10, "Editing the Base Configuration File" on page 73.

Appendix C. Replacing the Quiesce Check

This appendix describes how you can replace the quiesce check on any target.

What the Quiesce Check Does

Some change control operations check that the computer is not in use before the operations actually begin. This check is called the *quiesce check*, and, by default, checks that no users are logged on to the system. The results of the check are then returned to TME 10 Software Distribution. However, there may be occasions where the default quiesce check is not appropriate. For example, if users usually do not log off at the end of a work day, then the quiesce check always fails. A check that no processing power was being used is a more appropriate check.

Also, checking that no users are logged on does not prove that no processes are running that might interfere with the change control operation. For example, you may be attempting to uninstall a word processor on a computer while a background task (started by a user who is now logged off) is performing a batch of mail merges.

TME 10 Software Distribution enables you to address these kinds of scenarios by allowing the quiesce check to be replaced by a more appropriate check. Different targets on a network can each have different quiesce checks. This appendix describes how to replace the default TME 10 Software Distribution quiesce check with a check more suitable to the needs of your environment.

Implementing Quiesce Checks

The quiesce check on a target is performed by a script. This script is named `quiesced` and is stored in the directory with the other TME 10 Software Distribution scripts:

`<product_directory>script`

To replace the quiesce check, all you need to do is replace the script. The new script should:

- Not include any parameters
- Return 0 if the target is quiesced
- Return 2 if the target is not quiesced.

Example of a Check Script Used for Quiesce Checks

Figure 160 on page 254 shows an example of a quiesce script that performs the same function as the default script supplied with TME 10 Software Distribution, returning 0 if no users are on the system and returning 2 otherwise.

Example of a Check Script Used for Quiesce Checks

```
#
# Checks if the Target computer is quiesced or not depending
# on the number of logged-in users. The Target is deemed to
# be quiesced if no users are logged-in.
#
#
# Returns:
#
# 0 - the Target is quiesced
# 2 - the Target is not quiesced
#

numusers= who | wc -l          # Calculate the number of logged-in users
                                # by listing the users and counting the
                                # number of lines.

if test $numusers = 0 then     # If no users are logged in
    exit 0                     # Return 0, meaning 'quiesced'
else                           # Otherwise
    exit 2                     # Return 2, meaning 'not quiesced'
fi
```

Figure 160. Example of a check script used for quiesce checks

Appendix D. Writing Change Control Scripts

This appendix provides guidelines to help you write a change control script. It describes the parameters for scripts and how you can use them.

You can call change control scripts before and after most change control operations. The names of the scripts are contained within the software object. If no name is specified, no script is executed.

In addition to being named in the software object, scripts must be included as files to be installed unless they already exist on the target. The specification must refer to the complete path name where the script will be installed at the target.

Each script has a different set of parameters. These parameters, detailed in the next section of this appendix, are passed to it by the change control driver when it is executed.

A script produces a return code of 0 if it is successful. Any other value signifies that an error has occurred. If an error occurs, the entire change control request that is scheduled fails.

When scripts are being executed, any output generated to standard output is automatically redirected to a file called `request.out`. This file is stored in the work area. It is deleted before each new request.

The following sections describe the parameters that the driver passes to change control scripts when they are executed.

Creating Pre-Install and Post-Install Scripts

Use the following parameters for the pre-install and post-install scripts:

Install to Active Area

Whether the software object is being installed to the active or service area.
Valid entries are:

YES The installation is to the active area.

NO The installation is to the service area.

Service Subdirectory

Name of the service subdirectory. This is NUL if the installation is to the active area.

Removable

Whether the installation is removable. Valid entries are:

YES The installation is removable.

NO The installation is not removable.

DESIRED Try to make the installation removable. If it cannot be done, do not fail the installation.

Creating Pre-Accept and Post-Accept Scripts

Backup Subdirectory

Name of the backup subdirectory. This entry is NUL if the installation is not removable.

Creating Pre-Remove and Post-Remove Scripts

Use the following parameters for the pre-remove and post-remove scripts:

Request type

The type of request that gave rise to this event. Possible values are REMOVE, or INSTALL if the current removal is taking place because an install request failed to complete.

Action

The action to be taken by the remove may take one of these three values:

DELETE_SERVICE The installation should be deleted from the service subdirectory

RESTORE_SERVICE The removal should be performed in the service area

RESTORE_ACTIVE The removal should be performed in the active area.

Backup Subdirectory

The name of the backup subdirectory.

Service Subdirectory

The name of the service subdirectory. This is NUL if the action was RESTORE_ACTIVE.

Creating Pre-Accept and Post-Accept Scripts

Use the following parameters for the pre-accept and post-accept scripts:

Request type

This specifies the type of request that caused this event. Possible values are ACCEPT, or INSTALL if the current accept is taking place because of an install request with automatic acceptance specified.

Backup Subdirectory

The name of the backup subdirectory.

Service Subdirectory

The name of the service subdirectory. This is NUL if there is no service subdirectory.

Creating Pre-Uninstall and Post-Uninstall Scripts

Use the following parameters for the pre-uninstall and post-uninstall scripts:

Action

The action to be taken by the accept may take one of four values:

DELETE_SERVICE	The installation should be deleted from the service subdirectory
RESTORE_SERVICE	The uninstall operation should be performed in the service area
RESTORE_ACTIVE	The uninstall operation should be performed in the active area
DELSERV_RESTACT	The change file should be deleted from the service area and the software object should then be uninstalled from the active area.

Backup Subdirectory

The name of the backup subdirectory. This is NUL if the software object was not removable.

Service Subdirectory

The name of the service subdirectory. This is NUL if the action is RESTORE_ACTIVE.

Creating Pre-Activate Scripts

Use the following parameters for the pre-activate script:

Request type

This specifies the type of request that is being activated. Possible values are INSTALL, REMOVE, and UNINSTALL.

Service Subdirectory

The name of the service subdirectory.

Understanding Script Exit Codes

All pre- and post-requests, pre- and post-scripts, and procedures invoked by TME 10 Software Distribution return the following exit codes:

0	The script was run successfully.
1	This exit code is reserved for use by the system. It indicates that the shell returned an error.
2-50	The script returned an error.
51-100	The script was run successfully.

All other codes

This exit code is reserved for use by the system. It indicates that an error occurred.

Appendix E. Writing User Exits

General-Use Programming Interface

Before reading this chapter you should have an understanding of C language programming, and be able to compile and link applications under the operating system TME 10 Software Distribution is running under.

Overview

In certain areas of TME 10 Software Distribution, user exits are included. User exits are places in the code where you can add your own functions.

The C language source code for these empty functional areas is provided with both the TME 10 Software Distribution server and client. You can edit the code to make it perform the function that you require, and then compile it into a shared library from which it can be called by the program.

This appendix describes each user exit function and explains how to compile and link any changes that you make. It provides:

- A description of the files provided to enable you to write and compile user exit code.
- A description of the user exits, detailing when they are called.
- A description of the compilation methods to use to compile and link user exit code.

Files Provided

The following files are provided on both server and client, in the directory SRC on the CD-ROM, to assist you in writing user exit routines.

- Source files containing the empty user exit functions: `fndcx.c`, `fndcxcm.c`, `fndcxmo.c`, `fndssect.c`, and `fndsx.c`
- Make files, used when compiling and linking the user exit code: `fndcx.mak`, `fndsx.mak` and `fndss.mak`

Note that these files are provided on the CD-ROM and are not installed on your workstation as part of product installation.

User Exits

This section describes each of the user exit functions that TME 10 Software Distribution supports. The next section describes the mechanics of compiling and linking any changes that you make.

Writing User Exits

Header Files

Some of the user exits receive as parameters structures that are used in the TME 10 Software Distribution program. To enable you to use these structures, the necessary header files are supplied. These files contain comments describing the meanings of the fields in the structures.

Because these files are the header files used for the actual product, they contain the definitions of structures that you do not require. Ignore these extra structures.

The structures and fields of interest to you are mentioned in the description for each of the user exits. Do not change any fields unless you are advised that you can do so. IBM will not support problems due to use of unauthorized fields in user exits.

Generating Local File Names (ss_user_loc_name)

This user exit is used only on the server. It is called when TME 10 Software Distribution needs to store a file, but does not have a local name to use. This event occurs when a distribution is received or a software object is built and cataloged without being given a local name.

TME 10 Software Distribution generates a default local name by concatenating the label with the token \$(REPOSITORY). The default local name is then passed to this user exit, that can modify the name. Control is then returned to TME 10 Software Distribution, which uses the updated name to store the local file.

For software objects and plans transmitted over server-to-server connections, however, the user exits is always called before cataloging the file to allow the user to change the local file name used. For the other types of files, the user exit is called to allow the user to change the local file name corresponding to the file system ID specified.

The name is passed as a pointer to a character array. This file contains the local file name as a null terminated string in a 256-byte buffer. If you replace it with a longer string, make sure that you do not exceed the 256-byte limit.

This user exit is located in the source file `fndssext.c`, and is called `ss_user_loc_name`.

Notifying Requests (sx_server_request)

This user exit is used only on the server. It is called after a change control or distribution request has been submitted, and is passed a data structure containing the request data.

This user exit can be used, for example, to record that a request is scheduled for a target and then correlate the request with the next user exit (`sx_server_report`) when the current request is completed.

This user exit is supplied with a pointer to an `RR_INFO` (defined in `fndcx.h`) and to a `CX_USER_RESPONSE` structure (defined in `fndcx.h`).

The `RR_INFO` structure contains the request to be performed in the `type_data` field. The `CX_USER_RESPONSE` structure is currently not used. In the future it may be possible for the user exit to return a value to this structure to specify that the request should be postponed or canceled.

Notifying Completion (`sx_server_report`)

This user exit is used only on the server. It is called when TME 10 Software Distribution receives a report from a target, immediately before the report is deleted, and after the update of the status in the database. A data structure containing the report is passed to it.

This user exit is supplied with a pointer to an `RR_INFO` (defined in `fndcx.h`). The `RR_INFO` structure contains the request to be performed in the `type_data` field.

Modifying Local File Names (`cx_daca_filename`)

This user exit is present on all TME 10 Software Distribution workstations. It is called immediately before:

- Deleting a local file
- Storing a local file
- Sending a local file (after a send request)
- Sending a local file (in answer to a retrieve request)

The user exit is provided with the global and local names of the file. The local name has already undergone token substitution at the time that it is called. You can change the local file name, but not the label.

One use of user exit might be to reroute local files that should be stored in one directory to another directory, or to perform a more sophisticated form of token substitution than that provided.

The parameters to the user exit are the label held in a structure that is defined in `fndhdr.h`, and the local name stored as a null terminated string in a 256-byte character array. If you replace the local name with a longer string, make sure that you do not exceed the 256-byte limit.

The user exit is located in the source file `fndcx.c` and is called `cx_daca_filename`.

Previewing and Modifying Reports (`cx_daca_report`)

This user exit is present on all TME 10 Software Distribution workstations. It is called immediately before the target sends a report about change control or distribution activity back to the server. A data structure containing the report is passed to it.

You can use this user exit to keep local copies of reports that are sent to the server, so that the user of the target can see what change control and distribution events have taken place.

The user exit is supplied with a pointer to an `RR_INFO` structure that contains the report that will be sent to the server. This structure is defined in `fndcx.h`.

Writing User Exits

The `RR_INFO` structure contains the report that will be sent to the server in the `type_data` field. The `type` field contains the type of report held in the `type_data` field.

The user exit is located in the source file `fndcx.c`, and is called `cx_daca_report`.

Previewing Requests (`cx_daca_request`)

This user exit is present on all TME 10 Software Distribution workstations. It is called immediately before a target carries out a change control or distribution request, and passed a data structure containing a copy of the request.

You can use this user exit to provide a warning to all users of the system that a change control or distribution operation is about to take place.

The user exit is supplied with a pointer to an `RR_INFO` structure (defined in `fndcx.h`) and a `CX_USER_RESPONSE` structure (defined in `fndcx.h`).

The `RR_INFO` structure contains the request that is about to be performed in the `type_data` field.

The `CX_USER_RESPONSE` structure is currently not used. In the future it may be possible for the user exit to return a value in this structure to specify that the request should be postponed or canceled.

The user exit is located in the source file `fndcx.c` and is called `cx_daca_request`.

Exporting Directories (`sx_export_dir`)

This function is called by the server to export a list of directories to a given target. It is called before enqueueing a request to the target.

The function is defined as follows:

```
DC_USHORT  sx_export_dir (REMOTE_DIR *remote_dir,
                          TARGET_INFO *target_info,
                          DC_LONG    *errcode,
                          DC_USHORT  prot);
```

The `REMOTE_DIR` structure contains the following fields:

- Exported directory
- Mounted file system
- Mount options
- Export options

The values returned from the function are:

- 0 Directory successfully exported for the target. An informational message is logged.

- 4** Directory already exported for the target. An informational message is logged.
- 8** An error was encountered exporting the directory for the target. An error message is logged with the operating system error-code return in the output parameter *errcode*.

This user exit is located in the source file `fndsx.c`.



If this function is to be executed using the LAN Server services, the fields of the build profile concerning the build of a dynamic software object must be set as follows:

```
SERVER NAME:          <LAN Server Name> <Domain name>
EXPORTED DIRECTORY:   <directory to be exported>
MOUNTED FILE SYSTEM:  <drive used by the client to attach the server>
MOUNT OPTIONS:        <alias name of the exported directory>
```

At the client, the mount with LAN Requester results in a logon with userid equal to the TME 10 Software Distribution workstation name, followed by a net use command. LAN Server allows only one user to be logged on at a time. If a user is already logged on at the time of the mount, any disks already attached by the previous logon are detached by the new one.

Removing Exports (`sx_unexport_dir`)

This function is called by the server to remove a list of directories for a given list of targets. It is called when the request is completed.

The function is defined as follows:

```
DC_USHORT sx_unexport_dir (REMOTE_DIR *remote_dir,
                           TARGET_INFO *target_info,
                           DC_LONG *errcode,
                           DC_USHORT prot);
```

The values returned from the function are:

- 0** Directory successfully unexported for the target. An informational message is logged.
- 8** An error was encountered unexporting the directory for the target. An error message is logged with the operating system error-code return in the output parameter *errcode*.

This user exit is located in the source file `fndsx.c`.



If this function is to be executed using the LAN Server services, the fields of the build profile concerning the build of a dynamic software object must be set as follows:

SERVER NAME:	<LAN Server Name> <Domain name>
EXPORTED DIRECTORY:	<directory to be exported>
MOUNTED FILE SYSTEM:	<drive used by the client to attach the server>
MOUNT OPTIONS:	<alias name of the exported directory>

At the client, the mount with LAN Requester results in a logon with userid equal to the TME 10 Software Distribution workstation name, followed by a net use command. LAN Server allows only one user to be logged on at a time. If a user is already logged on at the time of the mount, any disks already attached by the previous logon are detached by the new one.

Mounting File Systems (cx_mount_fs)

This user exit is called by the agent to mount a list of file systems. It is called before executing the request.

The function is defined as follows:

```
DC_USHORT cx_mount_fs (REMOTE_DIR *remote_dir,  
                        DC_VOID **user_data,  
                        DC_USHORT prot);
```

The values returned from the function are:

- 0** File system successfully mounted. An informational message is logged.
- 4** File system already mounted. An error message is logged.
- 8** An error was encountered when mounting the file system. An error message is logged with the operating system error code return in the output parameter `errcode`.

This user exit is located in the source file `fndcxmo.c`.



If this function is to be executed using the LAN Server services, the fields of the build profile concerning the build of a dynamic software object must be set as follows:

SERVER NAME:	<LAN Server Name> <Domain name>
EXPORTED DIRECTORY:	<directory to be exported>
MOUNTED FILE SYSTEM:	<drive used by the client to attach the server>
MOUNT OPTIONS:	<alias name of the exported directory>

At the client, the mount with LAN Requester results in a logon with userid equal to the TME 10 Software Distribution workstation name, followed by a net use command. LAN Server allows only one user to be logged on at a time. If a user is already logged on at the time of the mount, any disks already attached by the previous logon are detached by the new one.

Unmounting File Systems (cx_unmount_fs)

This user exit is called by the agent to unmount a list of file systems. It is called when the request is completed.

The function is defined as follows:

```
DC_USHORT cx_unmount_fs (REMOTE_DIR *remote_dir,
                        DC_VOID **user_data,
                        DC_USHORT prot);
```

The values returned from the function are:

- 0** File system successfully unmounted. An informational message is logged.
- 8** An error was encountered while unmounting the file system. An error message is logged with the operating system error-code return in the output parameter `errcode`.

This user exit is located in the source file `fndcxmo.c`.



If this function is to be executed using the LAN Server services, the fields of the build profile concerning the build of a dynamic software object must be set as follows:

```
SERVER NAME:          <LAN Server Name> <Domain name>
EXPORTED DIRECTORY:   <directory to be exported>
MOUNTED FILE SYSTEM:  <drive used by the client to attach the server>
MOUNT OPTIONS:        <alias name of the exported directory>
```

At the client, the mount with LAN Requester results in a logon with userid equal to the TME 10 Software Distribution workstation name, followed by a net use command. LAN Server allows only one user to be logged on at a time. If a user is already logged on at the time of the mount, any disks already attached by the previous logon are detached by the new one.

Adding Target Parameters (sx_usrexit_trgcfg)

This user exit is called when a new target is defined. It is used to add any missing (optional) parameters to the target definition. The required fields must already be specified.

Writing User Exits

The function is defined as follows:

```
DC_SHORT sx_usrexit_trgcfg (const DC_USHORT      action
                           const USRX_TARGET_CONFIG *in_trg_config,
                           USRX_TARGET_CONFIG *out_trg_config);
/*****
```

Description:

This routine is called when a target configuration record is added or updated.

A user code can be specified to set the default value of a target configuration record.

The fields that the product considers are:

description	: descriptive information
target_os	: target operating system
password	: target password
lan_addr	: burned-in LAN address
cust	: customer name
contact	: contact name
contact_phone	: contact phone number
manager	: owning manager's name
manager_addr	: owning manager's address
num_hw_parm	: number of hardware parameters
hw_parm	: hardware parameter array
num_inst_parm	: number of installation parameters
inst_parm	: installation parameter array

Parameters:

action
ADD_CFG or UPD_CFG depending on command being executed

in_trg_config
input structure of target configuration record

out_trg_config
output structure of target configuration record in which
user-exit default values can be specified

Returns:

0 completed successfully

```
*****/
```

This user exit is located in the source file `fndsx.c`.

Adding User Parameters (sx_usrexit_usrcfg)

This user-exit is called when a new user is defined in order to add any missing (optional) fields. The mandatory fields have to be already specified.

The function is defined as follows:

```
DC_SHORT sx_usrexit_usrcfg (const DC_USHORT      action
                           const USRX_USER_DEF  *in_usr_def,
                           USRX_USER_DEF  *out_usr_def);
/*****
```

Description:

This routine is called when a user configuration record is added or updated.

A user code can be specified to set the default value of a user configuration record.

The fields that the product considers are:

description : descriptive information

Parameters:

action

ADD_CFG or UPD_CFG depending on command being executed

in_usr_def

input structure of user configuration record

out_usr_def

output structure of user configuration record in which user-exit default values can be specified

Returns:

0 completed successfully

```
*****/
```

This user exit is located in the source file fndsx.c.

Reporting Request Completion (**sx_server_report**)

This user exit is called when a report is returned to the server immediately after the change control status has been updated.

The function is defined as follows:

```
DC_VOID sx_server_report (REPORT_INFO *info);
/*****

Description:
  This routine receives the REPORT_INFO structure immediately after
  a report has been received by the server and saved in the DB.

Parameters:
  entry
    pointer to a REPORT_INFO structure.

Returns:
  none

*****/
```

This user exit is located in the source file `fndsx.c`.

Reporting Fileset Statuses and Products (**cx_vercm**)

This function is called by a TME 10 Software Distribution agent when it has to verify the status of filesets contained in a software object. It is run at a workstation where an agent product is running, and expresses the status for filesets as a TME 10 Software Distribution change management status.

For each fileset in the software object, the agent calls the `cx_vercm` user exit and creates a report in file `fndswcms`. If the status of a fileset does not coincide with the status of the software object, the status reported by `cx_vercm` is In error.

This user exit is located in the source file `fndcxcm.c`. It has the following format:

```
DC_SHORT cx_vercm (DC_CHAR      *cf_type,
                  DC_EXT_PRODUCT *product,
                  DC_EXT_FILESET *fileset,
                  DC_CHAR      *mask);
```

Product Information

Information related to products is also provided as input to this user exit. The same information is reported in the `fnswcms` file. It is supplied using the following format:

```
typedef struct dc_ext_product (DC_CHAR          tag[MAX_TAG_LEN];
                             DC_CHAR          revision[MAX_REVISION_LEN];
                             DC_CHAR          architecture[MAX_ARCHITECTURE_LEN];
                             DC_CHAR          vendor_tag[MAX_VENDOR_TAG_LEN];
                             DC_CHAR          title[MAX_PF_TITLE_LEN];
                             ) DC_EXT_PRODUCT;
```

Status Exchange Area

The area in the user exit used to exchange information about the status of a file has the following format. The agent provides the change management status of the software object to the procedure using the `status` field. The `cx_vercm` function determines the status of the file at the workstation and updates the `status` field.

```
typedef struct dc_ext_fileset (DC_CHAR          tag[MAX_TAG_LEN];
                             DC_CHAR          revision[MAX_REVISION_LEN];
                             DC_CHAR          title[MAX_PF_TITLE_LEN];
                             DC_CHAR          status;
                             DC_USHORT        fseterror;
                             ) DC_EXT_FILESET;
```

If the `fseterror` field contains the value `FS_NOT_DISCOVERED`, then `cx_vercm` reports the Not discovered status in the `fnswcms` file.

TME 10 Software Distribution Statuses

A status is represented by one byte, where:

- The four least significant bytes represent a change management status.
- The four most significant bytes represent a specific TME 10 Software Distribution status.

TME 10 Software Distribution statuses can be one of the following:

- Available
- Installed, removable, active
- Installed, not removable, active
- Installed, removable, inactive
- Installed, not removable, inactive
- Scheduled
- In error
- Removed, inactive
- Uninstalled, inactive

Writing User Exits

- Backlevel
- Discovered
- Distributed
- Distribution pending
- Authorized at target
- Authorized at server

Setting Secure Keys (cx_set_secure_key)

This user exit is included in the TME 10 Software Distribution base package. It can be called when software objects are built and the **secure package** attribute is set (see “Defining Data Security” on page 42).

The function calculates and associates a *secure key* with the software object. The user exit is defined as follows.

```
extern DC_VOID cx_set_secure_key( DC_ULONG  *crc_vector,
                                DC_USHORT  number_of_crcs,
                                DC_VOID     *secure_key,
                                DC_USHORT  secure_key_len);
/*****
```

Description:

This function is called by Software Distribution user interfaces to calculate, using a user-defined algorithm, the secure key to associate with a software object.

Parameters:

crc_vector: Pointer to the array containing the CRCs associated with all the objects in the software object (the CRC is 0 for directories and for objects to be deleted).

number_of_crcs: Number of elements in the vector.

secure_key: Buffer containing the calculated secure key to be checked at installation time (to be returned). Software Distribution assumes that if all the bytes are equal to \0, the buffer is empty and no secure key will be associated with the software object. This is the default.

secure_key_len: Length of the secure key (equal to 32 bytes)

Return Codes: None

```
*****/
```


Checking Secure Keys (cx_check_secure_key)

This user exit is included in the TME 10 Software Distribution client package. It checks the secure key associated with the software object using the cx_set_secure_key user exit to make sure that the file has not been manipulated. It can be used to:

- Install the software object only if it contains a secure key.
- Install the software object only if it contains a secure key, and information in the secure key semantics identifies the target where the software object was built (for example, if the first three characters of the key are 111).

The user exit is defined as follows.

```
extern DC_VOID cx_check_secure_key( DC_VOID  *secure_key,
                                   DC_USHORT secure_key_len,
                                   DC_USHORT *action);
/*****
```

Description:

This function is called by Software Distribution when performing an installation to check the secure key set in the software object by the cx_set_secure_key.

Parameters:

secure_key: Buffer containing the secure key to be check.
 secure_key_len: Length of the secure key (32 bytes).
 action: Possible values are:

- CM_INSTALLABLE
 This is the default. If this value is returned, the agent proceeds with the installation.
- CM_NO_INSTALLABLE
 If this value is returned, the agent halts the installation and reports a failure.

Returns: None

```
*****/
```

Checking CRC Numbers (cx_check_crcs)

This user exit is included in the TME 10 Software Distribution client package. It recalculates the secure key associated with the software object using the cx_set_secure_key user exit to make sure that it is the same. If it is not, the software object is not installed.

The user exit is defined as follows.

```
extern DC_VOID cx_check_crcs( DC_ULONG  *crc_vector,  
                             DC_USHORT  number_of_crcs,  
                             DC_VOID    *secure_key,  
                             DC_USHORT  secure_key_len,  
                             DC_USHORT  *action);  
/*****
```

Description:

This function is called by Software Distribution Client when performing an installation to recalculate the secure key set in the software object and compare it with the key set by the cx_set_secure_key when the software object was prepared.

Parameters:

crc_vector: Pointer to the array containing the CRCs associated with all the objects in the software object (the CRC is 0 for directories and for objects to be deleted).

number_of_crcs: Number of elements in the vector.

secure_key: Buffer containing the calculated secure key to be verified during the installation.

secure_key_len: Length of the secure key (32 bytes).

action: Possible values are:

- CM_INSTALLABLE
This is the default. Proceed with the installation.
- CM_NO_INSTALLABLE

If this value is returned, it means that the comparison between the secure key calculated when the software object was prepared and the secure key calculated during installation was not successful. In this case Software Distribution Client ends the installation and reports a failure.

Returns: None

```
*****/
```

Compilation Method

After compilation, the user exits are contained in three DLLs: FND CX.DLL, FND SX.DLL, and FND SS.DLL. Three make files, FND CX.MAK, FND SX.MAK, and FND SS.MAK, are provided for use in building the DLLs using the IBM C/C++ compiler.

The syntax of the command to obtain the DLL is:

```
nmake <makefilename>
```

Since the three DLLs obtained will reference the dynamic link library DDE4MBS.DLL provided by C++, after having built the three DLLs, you need to rename this DLL to FND4MBS.DLL, and also change the reference to this name in the FNDXXX.DLLs. You can do that with the utility DLLRNAME.EXE provided with IBM C/C++, as follows:

1 Copy the DDE4MBS.DLL to a working directory.

2 In this directory, enter the command:

```
DLLRNAME DDE4MBS.DLL DDE4MBS=FND4MBS
```

You will obtain a DLL named FND4MBS.DLL. Copy it into the subdirectory B'BIN of the product directory.

3 For each of the three DLLs you built, change the name by entering the command:

```
DLLRNAME FNDXXX.DLL DDE4MBS=FND4MBS
```

_____ End of General-Use Programming Interface _____

Appendix F. Application Definition File Considerations

Before you can use CID software preparation to prepare a software product for distribution, you must have an application definition file (ADF) for the product. CID-enabled products can also provide ADFs on their distribution media. You can also create your own ADFs.

If you decide to create an ADF, consider that you can choose from two approaches for providing responses during a CID installation. An ADF can specify that an existing response file is to be used, or it can specify that TME 10 Software Distribution is to generate the response file from information provided during CID software preparation.

Using an Existing Response File

If you choose this approach, the configuration of the product at the distribution client is done by using an external response file. Be sure that the response file is stored at the code server in the directory:

```
<drive>:\<CID_directory_path>\RSP\<product_name>
```

In Chapter 17, “Setting Up for CID Preparation” on page 215, the directory is:

```
C:\CID\RSP\<product_name>\
```

Generating the Response File

If you choose this approach, the product at the distribution client is configured during the CID software preparation task. TME 10 Software Distribution generates the response file based on the configuration values supplied during CID software preparation. The data can come from a dialog with the user doing CID software preparation, from the output of a user exit program, or from a file or database.

Figure 161 on page 276 shows an overview of the process of creating a software object starting from this kind of ADF.

Creating an Application Definition File

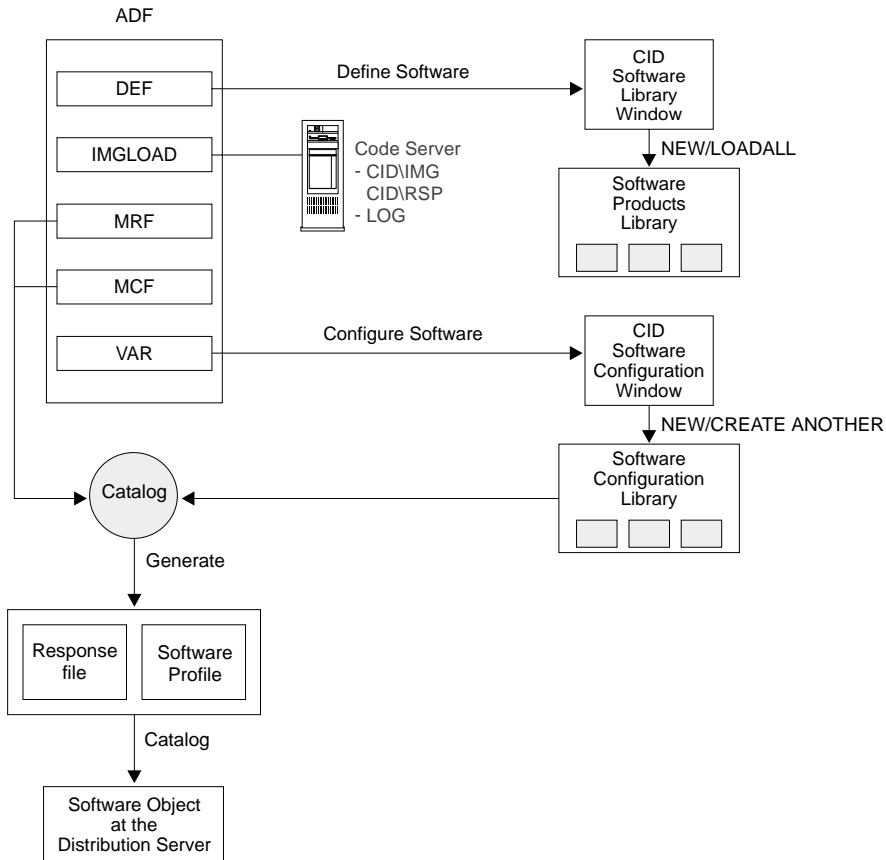


Figure 161. Generating a Software Object - the process

Creating an Application Definition File

To create your own application definition file for any CID-enabled product, follow one of these two approaches, depending on the type of application definition file you want to create:

1 Application definition file that points to an external response file

- a Create a file called <product name>.ADF.
- b Use the following example as a model; change the @DEF section and the installation command section as shown in the example. The lines to be changed are marked with "xxxxxx".

```

@DEF
* *****
*   Descr           is not specified here; it is the icon title
*
*   Name            is the short name of the product (max. 16 char)
*   BaseProd        is the short name of the Package that is the product name without
*                   explicit references to the version/release.
*   Release         is the version/release of the product (no dots allowed)
*   Level           is the maintenance Level of the product
*   Platform        is the required Operating System: OS2 or DOS
*   Category        is the type of application: OpSys, LANTrans, Distr, Application, CSD
*   Manufacturer    is the name of the company that produced the package
*   Language        is the NLS version of the product
*   DefCfgFile      is the file (*.NDI) that defines the default configurations
*   CfType          is the software object type (REF, CSD, or FIX)
* *****
      Description      "xxxxxxxxxxx"
      Name             xxxxxxxx (Up to 16 Characters)
      BaseProd         xxxxxxxx (Up to 16 Characters)
      Release          xxxxxxxx (Up to 16 Numeric Characters)
      Level            xxxxxxxx (Up to 16 Numeric Characters)
      Platform         xxxxxxxx (Up to 16 Characters)
      Category         Application
      Language         xxxxxx (Up to 16 Characters)
      Manufacturer     IBM

@ENDDF
@IMGLD
* *****
*   This section provides the information necessary to locate
*   product images on CD-ROM or diskette, upload the images to
*   the code server, add the product to the software library, and,
*   optionally, create and catalog a default configuration.
*
*   Prompt           is the message text to display to the user (for example, "Insert
*                   Diskette 1").
*   MarkerFile       is a file on the diskette; if it is missing, the diskette
*                   is the wrong one.
*   VolumeLabel      is the diskette label. Either MarkerFile or VolumeLabel
*                   is required.
*   SourcePath       is the path to the images in the image drive. They will be copied
*                   to the target drive under the same path.
*   Command          is the product upload program (XCOPY or another program).
*                   In the invocation, variables for source path and target
*                   directory may be used.
*   UserInteract     is YES if the upload program requests user data, NO otherwise
*   GoodRC           is the successful return code from the upload program. The default is 0.
* *****

@STEP<1>
      Prompt = "xxxxxxx"
      MarkerFile = xxxxxxxx

```

Creating an Application Definition File

```

        VolumeLabel = xxxxxxxx
        SourcePath = xxxxxxxx
        Command = XCOPY $T\CLIENT_W
        UserInteract = YES
        GoodRC = 0
    @ENDSTEP<1>
@ENDIMGLOAD
* *****
*               Parameter section
* This section lists all the parameters that are used during the process
* of generating response and change files; the values are collected
* when you configure this software
* *****
@VAR
* *****
* The following included file contains variables about the set-up of
* the code server
* *****
@INCLUDE CSREMOTE.VAR
* *****
* The following section contains the command to install and
* uninstall the product
* Use special symbols
*   $S      to specify the path containing the product images
*   $R      to specify the response file
*   $B      to specify the boot drive
*   $L1-5   to specify up to 5 log files
*   $L      to specify the first log file
* Each of them has a corresponding SDM_xxxxxx keyword in the
* SDMCMD.S.VAR that translates the command in the
* TME 10 Software Distribution format.
* Note:
* It is recommended to specify the drive or directory in which to install
* the software as a keyword of the response file rather than a parameter
* of the command line.
* *****
Section Commands
{
    InstallProgram
    {
        "XXXXX.EXE"
    }
    InstallParms
    {
        "/S:$S /R:$R /L1:$L1 /L2:$L2 /L3:$L3"
    }

    UninstallAllowed { 0 }
    UninstallProgram
    {
        "XXXXXXX.EXE"
    }
}
```



```

UninstallParms
{
  "/S:$S /R:$R /L1:$L1 /L4:$L4 /A:D"
}
}

*****
** INCLUDE here the configuration keywords that will be used in the
** model response file for the specified product
*****

*
@ENDVAR
* *****
*           Change File Profile skeleton section
*           -----
*  WARNING: do not change the lines between @MCF and @ENDMCF
* *****
@mcf
@INCLUDE CID.MCF
@ENDMCF
* *****
*           Response File skeleton section
*           -----
* Specify the name of the model response file in the include statement;
* the model R/F is a skeleton of a response file with place-holders
* instead of keyword value and conditional statements to include or exclude
* part of the file, depending on the value of some configuration keywords.
* *****
@mrf
@INCLUDE DUMMY.MRF
@ENDMRF

```

2 Application definition file for response file generation

- a Create files called <product name>.ADF, <product name>.MRF, and <product name>.VAR.

Refer to “Preparing an Application Definition File” in the TME 10 Software Distribution Information folder for detailed coding instructions.

The variable **UninstallAllowed** within an ADF can assume one of the following values:

- 0 The product cannot be uninstalled using CID.
- 1 The product can be uninstalled using CID

If you set the variable **UninstallAllowed** = 1 in the ADF of the product to allow uninstallation, during the software preparation task you must:

- Set the variable **Remote Response File** = **Yes** in the **Distribution** page of the configuration notebook
- Specify on the same page the name of the external response file and the subdirectory where the response file is stored.

Appendix G. Setting Environment Variables

To accomplish the tasks listed below, you must set the required environment variables.

To set an environment variable permanently, save it in the CONFIG.SYS file and reboot the workstation. For example, to set ACT_ON_TARGET to 1 permanently, set the following in the CONFIG.SYS file:

```
set ACT_ON_TARGET=1
```

Setting Compression or Decompression at the Local Server or at the Destination Target

Set the ACT_ON_TARGET environment variable to choose whether the transmitted file is to be compressed or decompressed at your local server or at the destination target.

Valid values and their meaning are the following:

- 0** The compression and decompression are performed at your local server.
- 1** The compression and decompression are performed at the destination target.

For targets that have TME 10 Software Distribution version prior to Version 3.1 installed, this variable can be set only to 0, which is the default value.

Improving the Performance of the Database Access Mechanism on OS/2 4.0

Set the CNDX_DEF_FREQ environment variable and the DISKCACHE=D,LW variable to increase the performance of the database by up to 70%. These variables affect the access mechanism.

You must set the variables in the CONFIG.SYS file as follows:

```
set CNDX_DEF_FREQ=YES  
set DISKCACHE=D,LW
```

The DISKCACHE=D,LW variable enables the LazyWrite feature of OS/2, which in the case of a power-off of the workstation or a power blackout, can preserve the integrity of the database. It is recommended to back up the <product_dir>\DB directory before distributing data.

Preventing the Loss of Icons When Performing CID Installations

Set the FNDACTDELAY environment variable to *seconds*, where *seconds* can be an unlimited numbers of seconds, to prevent the loss of icons when performing a CID installation. If you set this variable the system reboot is delayed for the specified number of seconds. If you set this variable, the icons will be correctly updated and OS/2 can refresh its buffer correctly because the system does not reboot immediately after the CID installation. If you do not set FNDACTDELAY, the system reboots immediately after the CID installation is completed.

Activating an Immediate Reboot

Set the FNDACTIMMEDIATE environment variable to a value to reboot the machine immediately when an activate request is received. If you set FNDACTIMMEDIATE, the TME 10 Software Distribution Activate pop-up window that prompts you to specify a delayed or immediate reboot no longer displays. Do not use this variable normally because it causes an immediate reboot.

Avoiding a Check of Diskette in the Drive

Set the FNDCHECKBOOT environment variable to a value so that the activate process does not perform a check that verifies if a diskette has been inserted in the drive.

Using TME 10 Software Distribution for OS/2 Client with the Windows NT Server Multiprocessor via NetBIOS

If you use the TME 10 Software Distribution for OS/2 Client via NetBIOS on an underpowered workstation, connected for example to a multiprocessor server, you may experience connection problems. To avoid these problems, set the following variable in the CONFIG.SYS file:

```
set FNDNBOS2 = YES
```

then restart the workstation.

Disabling a Pre-Request Script Process

Set the FNDACTREQUESTS environment variable to NO to disactivate the pre-requests script processing.

Collecting Product Trace Information

Set the following environment variables to collect product trace information:

FNDITRC

General. Sets the product trace for all the component

FNDITRC_CO

Common Routines. Sets the product trace for the common routines.

FNDITRC_DAS

Database Access Services. Sets the product trace for the database access services.

FNDITRC_GI

Graphical Interface. Sets the product trace for the graphical interface.

You can set a value from 0 to 6. Setting the value to 0 causes the maximum level of details to display. Setting the value to 6 the traces are not set. These variables will be read when the first task starts.

Showing Milliseconds in the Date and Time Field

Set the FNDLOGMSEC environment variable to a value to add milliseconds in the date and time format of the FNDLOG. Do not use the message log GUI when the FNDLOG file has the date and time in the millisecond format because the GUI cannot display the FNDLOG content. This variable must be set to provide better performance measurements only.

Disabling the User ID and Password

Set the FNDUSER and FNDPASSWORD environment variable to *userid* and *password* respectively if you do not want to enter the user ID and password either when you start the GUI or after you enter a command from the command line.

Setting the Full Path Name of a Trace File

Set the FNDUEX_TRC environment variable to define the full path name of a trace file. User exits use this file to append the trace data. This variable can be set for user exits only.

Setting a GUI Retry Time

Set the GI_RETRY_TIME environment variable to establish the timeout for displaying the GUI. Specify a value in seconds. The default value is 40.

Improving the Performance of the Target History Dialog

Set the HIDE_HIST_CM_PB environment variable to improve the performance of the Target History Dialog window. To cause the Target History Dialog window to be displayed faster, set HIDE_HIST_CM_PB to **yes**.

If you do this, the buttons for the CM functions are not displayed. To execute the CM functions, go to the Catalog window.

Setting a Server-to-Server Permanent Connection

Set the STS_PERM_CONN environment variable to establish a permanent server-to-server (STS) connection and to start a task for each connection.

The variable can be either YES, NO or a numeric value expressed in milliseconds. For example, if you set STS_PERM_CONNECTION to YES and the connection process is inactive, the STS connection is active for the time value specified in the STS_IDLE_TIME variable. The process checks every three seconds if there are resources to process.

If you set the STS_PERM_CONNECTION variable to a numeric value, for example 5000, the process checks if there are resources to process every 5000 milliseconds.

Setting Environment Variables

To save memory usage, if you do not have a constant traffic rate, do not set this variable.

Index

Numerics

3745 gateway 95

A

access keys 41
ACT_ON_TARGET 281
active client 5
active, software object status 36
adding target parameters, user exit 265
adding user parameters, user exit 267
ADF (application definition file)
 INF file 44
 requirement for 215
 use by upload images utility 226
administrator authorization profile 39
administrator, change control 9
AIX client
AIX, TME 10 Software Distribution for
 as a focal point 8
 as focal point 63
 configuring at 69
 configuring TME 10 Software Distribution for 59
 level required 52
 remote connection with 21
 response file keywords 243
 STS connection configuration file 69
 STS routing table 69
aliases for redirection 226
allocation failure retry time, SNA/DS configuration
 file 134
API trace file size, base configuration parameter 73
APPC
 configuration using SmartGuide 66
 connection with MVS 91
 Personal Communications configuration at
 client 118
 requirements 52
application definition file (ADF)
 INF file 44
 requirement for 215
 use by upload images utility 226
ASCII files, translation 34
attach manager 94
attended installation

attended installation (*continued*)
 distribution server 53
authorization profiles, user 39
authorizations, user 83
authorize, base configuration parameter 74
authorizing installation of software objects
 example 205
AUTOEXEC.BAT
 monitoring in a DiskCamera installation 181
automatic registration
 client 89
 definition 20
 keywords 89
 target 89
automatic target inventory, base configuration
 parameter 74
available, software object status 36

B

backlevel, software object status 36
backup area
 base configuration parameter 75
 directory for, server 58
 response file keyword 241
backup versions
 response file keyword to delete 240
base configuration file
 API trace file size 73
 authorize 74
 automatic target inventory 74
 automatic target registration 74
 backup area 75
 configuration 75
 connection window duration 75
 DACA idle time 75
 DACA retry time 75
 editing 73
 example 81
 file system 76
 format 73
 inventory program 76
 LAN authorization 76
 log file size 76
 machine type 76
 max connections 76

base configuration file (*continued*)

- max local targets 77
- max requests 77
- max server conns 77
- max server targets 77
- max SNA/DS routes 77
- max STS 77
- max STS routes 77
- max targets 78
- max user interfaces 78
- message log level 78
- purpose 73
- repository 80
- server 80
- service area 80
- STS idle time 80
- STS retry time 80
- target password authentication 80
- trace file size 81
- work area 81
- workstation name 81

books

- INF files 44
- PDF files 46
- PostScript 45
- TME 10 Software Distribution products xxi
- TME 10 Software Distribution, Version 3.1.5 for OS/2 44

builder authorization profile 40

builder, type of user 9

C

case sensitivity

- system name, server 57

catalog

- cleanup tool 151
- purpose 27
- recovery tool 151

catalog window 11, 12

CD-ROM

- requirement for 49
- SD4DOCPS directory 45
- SD4OS2\IMAGES directory 53

change control

- active status 36
- administrator 9
- available status 36
- backlevel status 36
- discovered status 36

change control (*continued*)

- distributed status 36
- distribution pending status 36
- in error status 36
- in introduction 3
- in progress status 36
- inactive status 36
- installed status 36
- not authorized status 36
- not installed status 36
- not removable status 36
- pristine installation 34
- reboot required status 36
- removable status 36
- scheduling 32

change control domain 4

change file

- See software object

checking secure keys 271

CID (configuration, installation, and distribution)

- adding products to software library 232
- code server setup 224
- customizing redirection mechanism 225
- directory structure 224
- environment 215
- installation method 27
- installation of TME 10 Software Distribution 237

CID Software Preparation window 13

cleanup tool, database 151

client, target type 86

clients

- authorizing installation by 205
- automatic registration 89
- fully disconnected 19
- installing an application on request 211
- mobile 18
- platforms supported 21
- role in network 4
- starting 149
- targets 17
- user interface only 9

CMIMAGE program 232

CNDX_DEF_FREQ 281

code server

- automatic setup 222
- command file 223
- copying images to 232
- setup 217
- upload images utility 226
- utility 217

- command line interface
 - help 45
 - INF file 44
 - starting 13
 - stopping 13
- commands, NetBIOS requirement, Warp 3.0x to Warp 4.0 51
- commands, NetBIOS requirement, Warp 4.5 52
- communication protocols
 - APPC, OS/2 configuration using SmartGuide 66
 - configuration, distribution server 57
 - response file keyword 241
 - software requirements 50
 - supported 22
- Communications Manager
- communications programs
- Communications Server 52
- components, TME 10 Software Distribution
 - disk requirements, distribution server 49
 - response file keyword to select 240
 - selecting during server installation 55
- compression, data 33
- compression, in software object 25
- CONFIG.SYS
 - increasing number of threads 166
 - monitoring in a DiskCamera installation 181
 - response file keyword to update 239
 - updating during server installation 54
- configuration
 - at the remote server site 69
 - base configuration parameter 75
 - Communications Server 91
 - distribution server 56
 - host connection 59
 - NetView DM for MVS 107
 - network drivers, distribution server 57
 - notebook, distribution server 56
 - Personal Communications 91
 - Personal Communications, at client 118
 - Personal Communications, for SNA/DS 113
 - Personal Communications, for STS 116
 - updating 71
- configuration file, SNA/DS
 - allocation failure retry time 134
 - editing 133
 - example 134
 - max CRMU 134
 - origin hop count 134
 - parameters 133
 - transmission hold time 134
- configuration file, STS
 - editing, STS 123
 - example, STS 124
 - origin hop count 124
 - parameters 123
- configuration notebook
 - distribution server 56
- configuration, installation, and distribution (CID)
 - adding products to software library 232
 - code server setup 224
 - customizing redirection mechanism 225
 - directory structure 224
 - environment 215
 - installation method 27
 - installation of TME 10 Software Distribution 237
- connection configuration file, SNA/DS
 - editing, SNA/DS 135
 - example for SNA/DS with APPC 136
 - fully qualified partner LU name 135
 - local LU alias name 135
 - mode name 135
 - next DSU 136
 - parameters 135
 - protocol 136
 - receive MU_ID time-out 136
 - remote server name 135
 - retry limit 136
 - send MU_ID time-out 136
 - transmission time-out 136
- connection configuration file, STS
 - editing 124
 - example for STS with APPC 125
 - example for STS with TCP/IP 125
 - example, AIX 69
 - protocol 125
 - remote server 69
 - remote server example 69
 - server name 125
 - type 125
- connection window duration, base configuration
 - parameter 75
- connectivity 22
- corequisites, in software object 25
- CRC numbers 271
- CSSETUP command file 223
- cx_check_crcs 271
- cx_check_secure_key 271
- cx_daca_filename 261
- cx_daca_report 261

- cx_daca_request 262
- cx_mount_fs 264
- cx_set_secure_key 270
- cx_unmount_fs 265
- cx_vercm 268

D

- DACA idle time, base configuration parameter 75
- DACA retry time, base configuration parameter 75
- DAK 41
- data access keys 41
- data compression 33
- data distribution
 - ASCII translation 34
 - compression 33
 - EBCDIC translation 34
 - translation 34
- data translation 34
- database cleanup tool 151
- database recovery tool 151
- DEFAULT.SID inventory dictionary file 247
- defining user authorization profiles 39
- deinstalling TME 10 Software Distribution 70
- delete pending request xxv
- dictionary file, inventory discovery 248
- directories for CID preparation 231
- directory specifications in software object 25
- discovered, software object status 36
- discovery, inventory
 - dictionary file 248
 - hardware filter 251
 - running 251
 - supplied program 247
 - tool 49
- disk requirements
 - distribution server 49
- disk space specifications in software object 25
- DiskCamera
 - basic scenario 181
 - drive for 182
 - files monitored by 181
 - Internet scenario 194
 - workplace objects monitored by 181
- DiskPicture
 - basic scenario 183
 - Internet scenario 199
- distributed, software object status 36
- distribution client
 - authorizing installation by 205

- distribution client (*continued*)
 - installing an application on request 211
 - starting 149
- distribution configuration notebook
 - distribution server 56
- distribution pending, software object status 36
- distribution server
 - APPC connections, OS/2 66
 - hardware prerequisites 49
 - increasing number of threads 166
 - installation 53
 - software prerequisites 49
 - starting 148
 - verifying installation results 166
- DLLs for user exits 273
- documentation component 49
- documentation folder 44
- domain 4
- drive
 - for DiskCamera 182
- DSKCAM.INI file 181
- dynamic groups 23
- dynamic software object
 - example of creation 172
 - purpose 28
- dynamic software objects 28

E

- EBCDIC files, translation 34
- editing remote connection files 133
- end user 10
- end user authorization profile 41
- environment variables
 - ACT_ON_TARGET 281
 - CNDX_DEF_FREQ 281
 - FNDACTDELAY 281
 - FNDACTIMMEDIATE 282
 - FNDCHECKBOOT 282
 - FNDLOGMSEC 283
 - FNDNBOS2 282
 - FNDPASSWORD 283
 - FNDREQUESTS 282
 - FNDUEX_TRC 283
 - FNDUSER 283
 - GI_RETRY_TIME 283
 - HIDE_HIST_CM_PB 283
 - STS_PERM_CONNECTION 283
- examples
 - adding products to software library 232

examples (*continued*)

- authorizing clients to install 205
- connection configuration file for APPC 125
- connection configuration file for TCP/IP 125
- copying product images 232
- discovered software package 249
- DiskCamera 181
- dynamic software object 172
- installation from a client 211
- installing from the Internet 194
- non-CID preparation, advanced 166
- non-CID preparation, basic 152
- pull installation 211
- push installation 161
- quiesce script 253
- remote directory 166
- SNA/DS configuration file 134
- STS configuration file 124
- STS routing table all protocols 130
- STS routing table for AIX 69
- STS routing table for APPC 129
- exit codes, scripts 257
- exporting directories, user exit 262

F

- fan out requests 20
- file system, base configuration parameter 76
- files provided for user exits 259
- filter, hardware inventory discovery 251
- filters for dynamic groups 23
- fix software object 26
- FNDACTDELAY 281
- FNDACTIMMEDIATE 282
- FNDACTREQUESTS 282
- FNDADMN 39
- FNDBLD 40
- FNDCHECKBOOT 282
- FNDHWINV hardware inventory file 250
- FNDINV inventory discovery program 247
- FNDITRC 282
- FNDITRC_CO 282
- FNDITRC_DAS 282
- FNDITRC_GI 282
 - refid=env.FNDITRC 282
 - refid=env.FNDITRC_CO 282
 - refid=env.FNDITRC_DAS 282
 - refid=env.FNDITRC_GI 282
- FNDLOG
 - verifying installation results 166

- FNDLOGMSEC 283
- FNDNBOS2 282
- FNDPASSWORD 283
- FNDSWINV software inventory file 249
- FNDTKINV installation parameters file 249
- FNDUEX_TRC 283
- FNDUSER 41, 283
- focal mode 84
- focal point target
 - configuring 63
 - NetView DM for MVS 22
 - use of 6
- fully disconnected client 19
- fully qualified partner LU name, SNA/DS connection configuration file 135

G

- gateway, 3745 95
- gateway, token-ring 97
- generic installation method 27
- GI_RETRY_TIME 283
- global name
 - See label
- grouping targets 23

H

- hardware inventory discovery 250
- hardware prerequisites
 - distribution server 49
 - in software object 25
- header files, user exits 260
- help window 11
- help, online 44
- HIDE_HIST_CM_PB 283
- holding operations 33
- host deletion of pending requests xxv

I

- IBM service, calling 46
- IBMLS, code server setup with 231
- IMAGES directory on CD-ROM 53
- images for CID preparation
 - copying to code server 232
 - upload utility 226
- in error, software object status 36
- in progress, software object status 36

- inactive, software object status 36
- INF files
 - ADF preparation 44
 - command line interface 44
 - displaying 44
 - message reference 44
 - pristine and migration scenarios 44
 - TME 10 Software Distribution information 44
 - TME 10 Software Distribution publications 44
 - TME 10 Software Distribution, Version 3.1.5 for OS/2 publications 44
- information folder 44
- INSTALL command 237
- installation through TME 10 Software Distribution
 - pull example 211
 - push example 161
 - verifying results 166
- installation, TME 10 Software Distribution
 - command 237
 - default directory, server 55
 - distribution server 53
 - interactive 53
 - overview 49
 - updating 70
 - with response file 237
- installed, software object status 36
- interface
 - command line, description 13
 - graphical user, description 10
 - GUI 10
 - starting, graphical 14
- intermediate node 20
- Internet
 - DiskCamera scenario 194
- inventory discovery
 - dictionary file 248
 - hardware filter 251
 - running 251
 - supplied program 247
 - tool 49
 - writing an inventory program 248
- inventory program, base configuration parameter 76
- IPX
 - configuration, distribution server 57
 - software requirements, Warp 3.0x to Warp 4.0 51
 - software requirements, Warp 4.5 52
 - supported 50

K

- keywords
 - response file 239
 - SNA/DS connection configuration file 135

L

- label 26
- LAN Adapter and Protocol Support (LAPS)
 - level required, Warp 3.0x to Warp 4.0 51
- LAN authorization, base configuration parameter 76
- LAN Server, code server setup with 231
- LANTRAN.LOG
- LAPS (LAN Adapter and Protocol Support)
 - level required, Warp 3.0x to Warp 4.0 51
- local control point profile
 - Personal Communications configuration for MVS connection 92
 - Personal Communications SNA/DS configuration 113
 - Personal Communications STS configuration 116
 - Personal Communications, client APPC configuration 119
- local logical unit
 - Personal Communications configuration for MVS connection 92
 - Personal Communications STS configuration 117
- local LU alias name, SNA/DS connection configuration file 135
- local LU profile
 - Personal Communications SNA/DS configuration 114
- local queues window 11
- local target
 - maximum number 77
- local target statuses, changes to xxvi
- local targets 17
- log, message
 - size, base configuration parameter 76
 - verifying installation results 166
- logical link profile
 - Personal Communications configuration for MVS connection 92
 - Personal Communications SNA/DS configuration 114
 - Personal Communications STS configuration 117
 - Personal Communications, client APPC configuration 119
- logical unit definitions

- logical unit definitions (*continued*)
 - 3745 connection 96
 - APPC connections 66
 - SDLC connection 100
 - token-ring gateway connection 98
- logical unit name
 - STS routing table 127
- logon mode table 103
- LU name 138

M

- machine type, base configuration parameter 76
- make files for user exits 273
- manager mode 84
- manuals
 - INF files 44
 - PDF files 46
 - PostScript 45
 - TME 10 Software Distribution products xxi
 - TME 10 Software Distribution, Version 3.1.5 for OS/2 44
- max connections, base configuration parameter 76
- max CRMU, SNA/DS configuration file 134
- max local targets, base configuration parameter 77
- max requests, base configuration parameter 77
- max server conns, base configuration parameter 77
- max server targets, base configuration parameter 77
- max SNADS routes, base configuration parameter 77
- max STS routes, base configuration parameter 77
- max STS, base configuration parameter 77
- max targets, base configuration parameter 78
- max user interfaces, base configuration file parameter 78
- message help 45
- message log
 - size, base configuration parameter 76
 - verifying installation results 166
- message log interface 12
- message log level, base configuration parameter 78
- message log window 11
- message reference online file 44
- migration scenarios 44
- minimum distribution priority 140
- mobile client 18
- mode name, SNA/DS connection configuration file 135
- mode profile
 - Personal Communications configuration for MVS connection 93
 - Personal Communications SNA/DS configuration 115

- mode profile (*continued*)
 - Personal Communications STS configuration 118
 - Personal Communications, client APPC configuration 120
- modes, target
 - client 17
 - mobile client 18
 - server 17
 - user interface only 19
- modifying local file names, user exit 261
- mounting file systems, user exit 264
- MPTS (Multi Protocol Transport Support)
 - requirement, Warp 3.0x to Warp 4.0 51
 - requirement, Warp 4.5 52
- Multi Protocol Transport Support (MPTS)
 - requirement, Warp 3.0x to Warp 4.0 51
 - requirement, Warp 4.5 52
- MVS, NetView Distribution Manager for
 - as a focal point 6
 - as focal point 63
 - configuring at 95
 - configuring Communications Server for 91
 - configuring Personal Communications for 91
 - configuring TME 10 Software Distribution for 59
 - defining targets for 94
 - level required 52
 - response file keywords 243

N

- name, of target 85
- name, system
 - response file keyword 241
 - server, specifying at installation 57
- names, NetBIOS requirement, Warp 3.0x to Warp 4.0 51
- names, NetBIOS requirement, Warp 4.5 52
- NCP
 - 3745 connection 96
 - SDLC connection 100
 - token-ring gateway connection 98
- NetBIOS
 - software requirements, Warp 3.0x to Warp 4.0 51
 - software requirements, Warp 4.5 52
- NetFinity Manager, requirement for inventory discovery 247
- NetFinity, limitation on coexistence 49
- NetView DM for MVS
 - as a focal point 6
 - as focal point 63

- NetView DM for MVS (*continued*)
 - configuring at 95
 - configuring Communications Server for 91
 - configuring Personal Communications for 91
 - configuring TME 10 Software Distribution for 59
 - defining targets for 94
 - level required 52
 - response file keywords 243
- NetView DM for NetWare
- NetView DM/2
- NetWare Requester, Warp 3.0x to Warp 4.0 51
- NetWare Requester, Warp 4.5 52
- network configurations
 - clients, definition 4
 - domain, definition 4
 - TME 10 Software Distribution in 3
- network drivers
 - configuration, distribution server 57
 - response file keyword 241
- network ID 127, 138
- new in this release xxiii
- next DSU, SNA/DS connection configuration file 136
- not authorized, software object status 36
- not installed, software object status 36
- not removable, software object status 36
- note definition file, Personal Communications 91
- notebooks
 - advanced software preparation 168
 - basic software preparation 154
 - code server setup 219
 - configuration, server 56
 - dynamic software objects 172
- notifying completion, user exit 261
- nvdm.cfg
 - See base configuration file

O

- origin hop count, SNA/DS configuration file 134
- origin hop count, STS configuration file 124
- OS/2
 - required on distribution server 49
- OS/2 client
 - files monitored by DiskCamera 181
 - Personal Communications configuration 118
 - supported 21
- OS/2 Warp
 - on distribution server 49
- OS2.INI
 - monitoring in a DiskCamera installation 181

- OS2INIT.CMD
 - monitoring in a DiskCamera installation 181
- OS2SYS.INI
 - monitoring in a DiskCamera installation 181
- overview 3

P

- partner local logical unit
 - Personal Communications STS configuration 117
- partner logical link profile
 - Personal Communications, client APPC configuration 120
- partner LU location profile
 - Personal Communications configuration for MVS connection 93
 - Personal Communications SNA/DS configuration 114
 - Personal Communications, client APPC configuration 120
- partner LU profile
 - Personal Communications configuration for MVS connection 93
 - Personal Communications SNA/DS configuration 114
- passive client 5
- PDF format books 46
- pending requests, delete xxv
- Personal Communications
 - APPC configuration using SmartGuide 66
 - configuring at a client 118
 - configuring for SNA/DS connection 113
 - configuring for STS connection 116
 - configuring to communicate with MVS 91
 - defaults, client APPC configuration 121
 - defaults, STS configuration 118
 - node definition file 91
 - requirement for 52
- physical unit definitions
 - 3745 connection 96
 - SDLC connection 100
 - token-ring gateway connection 98
- platforms supported xxiii
- platforms, client 21
- platforms, remote 21
- post-accept scripts 256
- post-install scripts 255
- post-remove scripts 256
- post-uninstall scripts 257

- PostScript publications 45
- pre-accept scripts 256
- pre-activate scripts 257
- pre-install scripts 255
- pre-remove scripts 256
- pre-uninstall scripts 257
- preparation site 20
- preparation, software
 - example, advanced non-CID 166
 - example, basic non-CID 152
 - examples, DiskCamera 181
- prerequisites, hardware and software
 - hardware, distribution server 49
 - host communication 52
 - software, distribution server 49
- prerequisites, in software object 25
- previewing and modifying reports, user exit 261
- pristine installation 34
- pristine installation scenarios, new xxiii
- pristine scenarios 44
- processor requirements
 - distribution server 49
- products, in software object 25
- profiles, user authorization 39
- programs, in software object 25
- protocol, SNA/DS connection configuration file 136
- protocol, STS connection configuration file 125
- protocols
 - APPC, OS/2 configuration using SmartGuide 66
 - configuration, distribution server 57
 - response file keyword 241
 - SNA/DS connection configuration file 136
 - software requirements 50
 - supported 22
- publications
 - INF files 44
 - PDF files 46
 - PostScript 45
 - TME 10 Software Distribution products xxi
 - TME 10 Software Distribution, Version 3.1.5 for OS/2 44
- pull mode 84
 - definition 17
 - example 211
- push mode 84
 - definition 17
 - example 161

Q

- quiesce check 253

R

- RAM requirements
 - distribution server 49
- READ.ME file 44
- reboot required, software object status 36
- receive MU_ID time-out, SNA/DS connection configuration file 136
- reconfiguring
 - server 70
- reconfiguring TME 10 Software Distribution 71
- recovery tool 151
- recursive operations, scheduling 33
- redirected drive
 - response file installation from 237
- redirection software 216
- refresh software object 26
- registration, automatic
 - base configuration parameter 74
- Registry
 - monitoring in a DiskCamera installation 181
- remote directory, example 166
- remote platforms supported 21
- remote queues window 11
- remote server name, SNA/DS connection configuration file 135
- remote source software object 25
- remote targets 19
- removable installation
 - backup area for, server 58
- removable, software object status 36
- removing directories, user exit 263
- reporting fileset status and products, user exit 268
- reporting request completion, user exit 268
- repository
 - base configuration parameter 80
 - directory for, server 59
- requests
 - maximum number 77
- requests window 11
- requirements
 - hardware, distribution server 49
 - host communication 52
 - software, distribution server 49
- response files
 - at code server 224

- response files (*continued*)
 - directory for 224
 - in software object 25
 - installing TME 10 Software Distribution with 237
 - keywords 239
 - layout 239
- retry limit, SNA/DS connection configuration file 136
- root user
 - logging on as 65
 - password 65
 - security 39
- routing table, remote server 69
- routing table, SNA/DS
 - configuring 137
 - LU name 138
 - network ID 138
 - planning 138
 - server name 138
 - SNA/DS address 138
 - target address 138
- routing table, STS
 - defining routes 128
 - editing 126
 - example, all protocols 130
 - example, APPC 129
 - LU name 127
 - network ID 127
 - planning 127
 - server name 127
 - target address 127
 - target address, STS 127

S

- samples
 - adding products to software library 232
 - authorizing clients to install 205
 - connection configuration file for APPC 125
 - connection configuration file for TCP/IP 125
 - copying product images 232
 - discovered software package 249
 - DiskCamera 181
 - dynamic software object 172
 - installation from a client 211
 - installing from the Internet 194
 - non-CID preparation, advanced 166
 - non-CID preparation, basic 152
 - pull installation 211
 - push installation 161
 - quiesce script 253

- samples (*continued*)
 - remote directory 166
 - SNA/DS configuration file 134
 - STS configuration file 124
 - STS routing table for AIX 69
 - STS routing table for all protocols 130
 - STS routing table for APPC 129
- scheduling facilities
 - change control operations 32
- scripts
 - exit codes 257
 - in software object 25
 - post-accept 256
 - post-install 255
 - post-remove 256
 - post-uninstall 257
 - pre-accept 256
 - pre-activate scripts 257
 - pre-install 255
 - pre-remove 256
 - pre-uninstall 257
 - writing 255
- SD4DOCPS directory 45
- SD4OS2 directory on CD-ROM
 - IMAGES subdirectory 53
 - server installation 53
- SDLC connection 100
- secure key 42
- secure package 42
- security 39
- send MU_ID time-out, SNA/DS connection configuration file 136
- server target 19
- server, code
 - automatic setup 222
 - copying images to 232
 - non-LAN Server 224
 - setup 217
 - upload images utility 226
 - utility 217
- server, distribution
 - base configuration parameter 80
 - configuring 73
 - hardware prerequisites 49
 - increasing number of threads 166
 - installation 53
 - name, SNA/DS connection configuration file 125
 - name, SNA/DS routing table 138
 - name, STS routing table 127
 - publications xxi

- server, distribution (*continued*)
 - software prerequisites 49
 - starting 148
 - verifying installation results 166
- server, target type 86
- service area
 - base configuration parameter 80
 - directory for, server 58
 - response file keyword 241
- service, IBM 46
- sessions, NetBIOS requirement, Warp 3.0x to Warp 4.0 51
- sessions, NetBIOS requirement, Warp 4.5 52
- setting secure keys 270
- single node
 - as a preparation site 9
 - purposes 20
- single, target type 86
- SmartGuide, APPC configuration 66
- SNA
 - configuration using SmartGuide 66
 - configuration, distribution server 57
 - connection configuration file, editing 135
 - requirements 52
- SNA server TP
 - Personal Communications SNA/DS configuration 115
 - Personal Communications, client APPC configuration 121
 - Personal Communications, STS configuration 118
- SNA/DS address 138
- SNA/DS configuration file
 - allocation failure retry time 134
 - editing 133
 - example 134
 - max CRMU 134
 - origin hop count 134
 - parameters 133
 - transmission hold time 134
- SNA/DS connection
 - Personal Communications configuration 113
- software
 - Software Distribution for Windows 2000, remote connection with 22
 - Software Distribution for Windows NT, remote connection with 22
 - software installation
 - on pristine workstations 34
 - scheduling 32
 - software inventory discovery 249
 - software object
 - CID 27
 - compression 25
 - contents of 25
 - corequisites 25
 - description 25
 - directory specifications in 25
 - disk space specification 25
 - dynamic 28
 - fix 26
 - generic 27
 - labels 26
 - naming 26
 - platform specific 26
 - prerequisites 25
 - products in 25
 - programs in 25
 - refresh 26
 - remote source 25
 - response files in 25
 - scripts in 25
 - security 42
 - status 36
 - update 26
 - Software Object Profiles window 12
 - software preparation
 - example, advanced non-CID 166
 - example, basic non-CID 152
 - examples, DiskCamera 181
 - software preparation interface 12
 - Software Preparation window 12
 - software prerequisites
 - distribution server 49
 - host communication 52
 - in software object 25
 - Software profile - create another window 12
 - specifications, software object 25
 - SPX
 - configuration, distribution server 57
 - software requirements, Warp 3.0x to Warp 4.0 51
 - software requirements, Warp 4.5 52
 - supported 50
 - ss_user_loc_name 260
 - starting graphical interface 14
 - starting the command line interface 13
 - starting TME 10 Software Distribution 13
 - STARTUP.CMD
 - monitoring in a DiskCamera installation 181

- static software objects 28
- stattg command, changes to statuses xxvi
- stopping the command line interface 13
- stopping TME 10 Software Distribution 16
- STS configuration file
 - editing 123
 - example 124
 - origin hop count 124
 - parameters 123
- STS connection
 - connection configuration file, editing 124
 - editing remote connection files 123
 - Personal Communications configuration 116
- STS connection configuration file
 - editing 124
 - example for STS with APPC 125
 - example for STS with TCP/IP 125
 - protocol 125
 - remote server example 69
 - server name 125
 - type 125
- STS idle time, base configuration parameter 80
- STS retry time, base configuration parameter 80
- STS routing table
 - defining routes 128
 - editing 126
 - example, all protocols 130
 - example, APPC 129
 - LU name 127
 - network ID 127
 - planning 127
 - server name 127
 - target address 127
 - target address, STS 127
- STS_PERM_CONNECTION 283
- sx_export_dir 262
- sx_server_report
 - notifying completion 261
 - reporting request completion 268
- sx_server_request 260
- sx_unexport_dir 263
- sx_usrexit_trgcfg 265
- sx_usrexit_usrcfg 267
- system name
 - response file keyword 241
 - server, specifying at installation 57
- SystemView/400

T

- TAK 41
- target address 138
- target password authentication, base configuration parameter 80
- targets
 - access keys 41
 - automatic registration 20
 - dynamic groups 23
 - for communication with MVS 94
 - grouping 23
 - holding operations at 33
 - intermediate node 20
 - local 5
 - maximum concurrently active 24
 - maximum number, local 77
 - maximum number, total 78
 - pristine installations on 34
 - remote 5
 - server 19
 - single node 20
 - user access to 41
- targets window 11
- TCP/IP
 - configuration for communication with a remote server 68
 - configuration, distribution server 57
 - CSD level required, Warp 3.0x to Warp 4.0 51
 - CSD level required, Warp 4.5 52
 - supported 50
- threads, increasing number of 166
- TME 10 Software Distribution for AIX
 - as a focal point 8
 - as focal point 63
 - configuring at 69
 - configuring TME 10 Software Distribution for 59
 - level required 52
 - remote connection with 21
 - response file keywords 243
 - STS connection configuration file 69
 - STS routing table 69
- token-ring gateway 97
- trace
 - file size, base configuration parameter 81
- tracking transmission requests
- transaction program
 - inbound 93
 - outbound 94
- transaction programs, APPC 66

- translation, data 34
- transmission hold time, SNA/DS configuration file 134
- transmission protocols
 - configuration, distribution server 57
 - response file keyword 241
 - SNA/DS connection configuration file 136
 - software requirements 50
 - supported 22
- transmission time-out, SNA/DS connection configuration file 136
- type, target 86

U

- UI only targets 87
- unattended installation
 - overview 32
 - TME 10 Software Distribution server 237
- uninstallation
 - distribution server 70
- unmounting file systems, user exit 265
- update software object 26
- upgrading
- upload software images utility 226
- user exits 259
 - adding target parameters 265
 - adding user parameters 267
 - checking CRC numbers 271
 - checking secure keys 271
 - compilation methods 273
 - cx_check_crcs 271
 - cx_check_secure_key 271
 - cx_daca_filename 261
 - cx_daca_report 261
 - cx_daca_request 262
 - cx_mount_fs 264
 - cx_set_secure_key 270
 - cx_unmount_fs 265
 - cx_vercm 268
 - DLLs 273
 - exporting directories 262
 - functions supported by 259
 - generating local file names 260
 - header files 260
 - make files 273
 - mounting file systems 264
 - notifying requests 260
 - previewing and modifying reports 261
 - previewing requests, user exit 262
 - removing exports 263

- user exits (*continued*)
 - reporting fileset status and products 268
 - reporting request completion 268
 - setting secure keys 270
 - ss_user_loc_name 260
 - sx_export_dir 262
 - sx_server_report, notifying completion 261
 - sx_server_report, request completion 268
 - sx_server_request 260
 - sx_unexport_dir 263
 - sx_usrexit_trgcfg 265
 - sx_usrexit_usrcfg 267
 - unmounting file systems 265
 - writing 259
- user interface only client 19
- user interface only targets 87
- users
 - administrator 39
 - authorization profiles 39
 - authorizations 83
 - builder 9
 - builders 40
 - change control administrator 9
 - data access keys 41
 - defining 39
 - end user 10
 - root 39
 - root, logging on as 65
 - root, password 65
 - target access keys 41
 - users belonging to FNDUSER 41
- utility programs
 - code server setup 217
 - database cleanup 151
 - database recovery 151
 - to copy product images 232
 - upload software images 226

V

- variables
 - customizing 190
- VIEW command 44
- VTAM
 - 3745 connection 97
 - SDLC connection 101
 - token-ring gateway connection 98

W

- Warp, OS/2
 - on distribution server 49
- what's new xxiii
- Windows 2000 clients
- Windows 3.1 clients
 - files monitored by DiskCamera 181
 - supported 21
- Windows 95 clients
 - files monitored by DiskCamera 181
 - supported 21
- Windows 98 clients
- Windows NT clients
 - files monitored by DiskCamera 181
 - supported 21
- windows, graphical user interface
 - catalog 11
 - CID Software Preparation 13
 - help 11
 - local queues 11
 - message log 12
 - requests 11
 - Software Object Profiles 12
 - Software Preparation 12
 - Software profile - create another 12
 - targets 11
- work area
 - base configuration parameter 81
 - directory for, server 59
- workplace objects, monitored by DiskCamera 181
- workstation name, base configuration parameter 81
- writing change control scripts 255

X

- XCOPY, to copy product images 232

Communicating Your Comments to IBM

TME® 10 Software Distribution
for OS/2®
Quick Beginnings
Version 3.1.5
Publication No. SH19-4334-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you prefer to send comments by mail, use the reader's comment form (RCF) at the back of this book. If you wish, you can give the RCF to the local branch office or IBM representative for postage-paid mailing.

If you prefer to send comments by fax, use this number, which is in Italy:

39+06+596+62077

If you prefer to send comments electronically, use this network ID:

ROMERCF at VNET.IBM.COM

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Help us help you!

**TME® 10 Software Distribution
for OS/2®
Quick Beginnings
Version 3.1.5**

Publication No. SH19-4334-02

We hope you find this publication useful, readable, and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your task	<input type="checkbox"/>	<input type="checkbox"/>

Specific comments, problems, or suggestions:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name	Address
------	---------

Company or Organization	
-------------------------	--

Phone No.	
-----------	--

Help us help you!
SH19-4334-02



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

License Use Management Information Development
Rome Tivoli Laboratory
IBM Italia S.p.A.
Via Sciangai, 53
00144 Rome
Italy

Fold and Tape

Please do not staple

Fold and Tape

SH19-4334-02

Cut or Fold
Along Line



Program Number: 5639-B06

Distributed Electronically for Customer Printing