

TME® 10 Software Distribution
for Windows NT®



Quick Beginnings

Version 3.1.5

TME® 10 Software Distribution
for Windows NT®



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, SH19-4335-02

This edition applies to Version 3.1.5 of TME 10 Software Distribution for Windows NT, program number 5639-B01, 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 International Business Machines 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	10
Message Log Interface	11
Software Preparation Interface	12
Command Line Interface	12
Starting and Stopping the Command Line Interface	13
Starting TME 10 Software Distribution	13
Starting the Graphical Interfaces	14
Stopping TME 10 Software Distribution	15
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
TME 10 Software Distribution Server Prerequisites	49
Hardware	49
Software	50
Communication Protocols	50
Protocols to Connect to Another Server	50
Protocols to Connect to a Client	51
Protocols to connect to a focal point	51
Prerequisites on the Host System	51

Chapter 9. Installing and Configuring the TME 10 Software Distribution Server for Windows NT or 2000	53
Installing TME 10 Software Distribution for Windows NT or Windows 2000 in Attended Mode	53
Changing Protocols after the Installation	61
Customizing the Server for NetBIOS	61
Customizing the Client for NetBIOS	64
Customizing the Server for IPX	64
Customizing the Client for IPX	65
Upgrading TME 10 Software Distribution for Windows NT	66
Using a Response File to Install a TME 10 Software Distribution Server	70
Uninstalling a TME 10 Software Distribution Server or Windows NT or Windows 2000	70
Removing Settings if the Uninstall Process Fails	72

Chapter 10. Editing the Base Configuration File	75
Base Configuration File Parameters	75
Changing TCP/IP Ports	85
Changing the TME 10 Software Distribution Server Name	86
Setting the LAN Authorization Level	86

Chapter 11. Defining Users and Targets	89
Defining Users	89
Defining Targets	89
Automatic Client Registration	95

Chapter 12. Configuring STS Remote Connection Files	97
Editing STS Connection Configuration Files	97

STS Configuration File	97
STS Connection Configuration File	98
Editing the STS Routing Table	99
Planning a Simple Routing Table	100
Planning a Complex Routing Table	100
Determining Destination Addresses	100
Defining Routes	101
Defining STS Targets	103
Chapter 13. Configuring SNA/DS Remote Connection Files	105
Editing Connection Configuration Files	105
SNA/DS Configuration File	105
SNA/DS Connection Configuration Files	107
Editing the SNA/DS Routing Table	110
Planning a Simple Routing Table	111
Planning a Complex Routing Table	111
Determining Destination Addresses	111
Defining Routes	112
Chapter 14. Configuring Microsoft SNA Server	117
Prerequisites	117
General Assumptions	117
Configuring SNA Server Parameters	117
Configuring LU Properties	120
Configuring APPC Modes	124
Configuring a Downstream Connection	128
Configuring Remote LU Properties	134
Creating CPI-C Information Records	138
Configuring the Local Transaction Program	142
Configuring the Local Transaction Program for the Send Function	143
Configuring the Local Transaction Program for the Retrieve Function	144
Chapter 15. Configuring IBM Communications Server	145
Prerequisite	145
General Assumptions	145
Configuring IBM Communications Server Parameters	145
Defining the Node	147
Defining a LAN Device	149
Defining a LAN Connection	152
Defining a Partner LU 6.2	155
Defining a Mode	157
Defining a Local LU 6.2	158
Defining CPI-C Side Information for the Receive Function	159
Defining CPI-C Side Information for the Send Function	161
Defining a Transaction Program for the Send Function	163
Defining a Transaction Program for the Receive Function	165
Saving the Configuration File	167
Starting the Node	168

Establishing an APPC Connection between the TME 10 Software Distribution for Windows NT Server and Another TME 10 Software Distribution, Version 3.1.5 Server	171
Defining the Windows NT Node at NetView DM for MVS Using GIX	171
Chapter 16. Configuring IBM Personal Communications	173
Prerequisite	173
General Assumptions	173
Configuring IBM Personal Communications Parameters	173
Defining the Node	175
Defining a LAN Device	177
Defining a LAN Connection	180
Defining a Partner LU 6.2	183
Defining a Mode	185
Defining a Local LU 6.2	186
Defining CPI-C Side Information for the Receive Function	187
Defining CPI-C Side Information for the Send Function	189
Saving the Configuration File	191
Configuring the Local Transaction Program	191
Configuring the Local Transaction Program for the Send Function	192
Configuring the Local Transaction Program for the Retrieve Function	192
Starting the Node	193
Establishing an APPC Connection between the TME 10 Software Distribution for Windows NT Server and Another TME 10 Software Distribution, Version 3.1.5 Server	196
Defining the Windows NT Node at NetView DM for MVS Using GIX	196
Using the Personal Communications GUIs with TME 10 Software Distribution	196

Part 3. TME 10 Software Distribution Scenarios 199

Chapter 17. Software Distribution Scenarios	201
The Software Distribution Environment	201
Distribution Database Recovery Tool	201
TME 10 Software Distribution Database Cleanup Tool	202
Scenario 1: Preparing an Application in Basic Mode	202
Scenario 2: Preparing an Application in Advanced Mode	211
Adding a Software Object Profile to the Software Library With Remote Directory	211
Adding a Dynamic Software Object to the Software Library	217
Scenario 3: Replicating an Installation with DiskCamera	227
Taking a “Before” Picture	227
Installing the Application at the Preparation Site	230
Taking an “After” Picture	231
Cataloging the Software Object	238

Part 4. Appendixes 241

Appendix A. Implementing Inventory Discovery	243
---	------------

OS/2 and Windows Inventory Discovery Program	243
Writing an Inventory Discovery Program	244
Mapping to the TME 10 Software Distribution Inventory Files	245
Installation Parameters Inventory File	245
Software Inventory File	245
Hardware Inventory File	246
Defining a Filter for Hardware Keywords	247
Running Inventory Discovery	247
Routing Inventory Information to Upstream Servers	248
Appendix B. Replacing the Quiesce Check	249
What the Quiesce Check Does	249
Implementing Quiesce Checks	249
Example of a Check Script Used for Quiesce Checks	249
Appendix C. Writing Change Control Scripts	251
Creating Pre-Install and Post-Install Scripts	251
Creating Pre-Remove and Post-Remove Scripts	252
Creating Pre-Accept and Post-Accept Scripts	252
Creating Pre-Uninstall and Post-Uninstall Scripts	253
Creating Pre-Activate Scripts	253
Understanding Script Exit Codes	253
Appendix D. Writing User Exits	255
Overview	255
Files Provided	255
User Exits	255
Header Files	256
Generating Local File Names (ss_user_loc_name)	256
Notifying Requests (sx_server_request)	256
Notifying Completion (sx_server_report)	257
Modifying Local File Names (cx_daca_filename)	257
Previewing and Modifying Reports (cx_daca_report)	257
Previewing Requests (cx_daca_request)	258
Exporting Directories (sx_export_dir)	258
Removing Exports (sx_unexport_dir)	259
Mounting File Systems (cx_mount_fs)	259
Unmounting File Systems (cx_unmount_fs)	260
Adding Target Parameters (sx_usrexit_trgcfg)	260
Adding User Parameters (sx_usrexit_usrcfg)	262
Reporting Request Completion (sx_server_report)	263
Reporting Fileset Statuses and Products (cx_vercm)	263
Product Information	264
Status Exchange Area	264
TME 10 Software Distribution Statuses	264
Setting Secure Keys (cx_set_secure_key)	265
Checking Secure Keys (cx_check_secure_key)	266
Checking CRC Numbers (cx_check_crcs)	266

Compilation Method	268
Appendix E. Setting Environment Variables	269
Setting Compression or Decompression at the Local Server or at the Destination	
Target	269
Disabling a Pre-Request Script Process	269
Collecting Product Trace Information	269
Showing Milliseconds in the Date and Time Field	270
Disabling the User ID and Password	270
Setting the Full Path Name of a Trace File	270
Setting a GUI Retry Time	270
Improving the Performance of the Target History Dialog	270
Setting a Server-to-Server Permanent Connection	271
Shutting Down the Product Using the FNDBOOT.cmd Command	271
Enabling Thread Termination During the Closure of the NVDMGI.exe Process	271
Hiding Black Windows Executing Script Files	271
Disabling the Shutdown Pop-up Menu During an Unattended Uninstallation	271
Index	273

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 Windows NT folder	13
6.	Logging on to TME 10 Software Distribution	14
7.	Distribution Server documentation - icon view	43
8.	Windows NT or Windows 2000 server - installation welcome window	54
9.	Windows NT or Windows 2000 server - select installation window	54
10.	Windows NT or Windows 2000 server - select components window	55
11.	Choose directory window	56
12.	Available disk space window	56
13.	Windows NT or Windows 2000 server - select program folder window	57
14.	Windows NT or Windows 2000 server - select protocol window	57
15.	Windows NT or Windows 2000 server - server configuration window	58
16.	Windows NT or Windows 2000 server - ServerName configuration window	59
17.	Windows NT or Windows 2000 server - user name configuration window	59
18.	Windows NT or Windows 2000 server - start product selection window	60
19.	Windows NT or Windows 2000 server - setup complete window	61
20.	Windows NT or Windows 2000 network settings window	62
21.	NetBIOS interface window	62
22.	Example of the NetBIOS interface window	63
23.	Example of the NetBIOS interface window	63
24.	Third example of the NetBIOS interface window	63
25.	Fourth example of the NetBIOS Interface window	64
26.	Windows NT server - installation welcome window	66
27.	Windows NT server - select client or server window	67
28.	Windows NT server - reinstall selection window	68
29.	Windows NT server - select components window	69
30.	Windows NT or Windows 2000 server - uninstall icon	71
31.	Windows NT or Windows 2000 server - uninstallation panel	71
32.	Example of a TME 10 Software Distribution Server Base Configuration file	85
33.	Example of an STS configuration file	98
34.	Example of an STS connection configuration file for TCP/IP	99
35.	Example of an STS routing table for TCP/IP routes	103
36.	Example of an SNA/DS configuration file	107
37.	Example of an SNA/DS connection configuration file for APPC	110
38.	Example of an SNA/DS routing table for APPC routes	116
39.	Example of an SNA/DS routing table for both APPC and TCP/IP protocols	116
40.	Microsoft SNA Server Manager	118
41.	Selecting SNA service	119
42.	Configuring network and control point names	120
43.	Selecting local APPC LUs	121
44.	Selecting an LU name	122
45.	Configuring APPC LU properties: general page	123

46.	Configuring APPC LU properties: advanced page	124
47.	Selecting APPC modes	125
48.	Configuring APPC modes: general page	126
49.	Configuring APPC modes: limits page	127
50.	Configuring APPC modes: characteristics page	128
51.	Selecting a downstream connection	129
52.	Configuring connection properties: general page	130
53.	Configuring connection properties: address page	131
54.	Configuring connection properties: system identification page	132
55.	Configuring APPC modes: 802.2 DLC page	134
56.	Configuring remote LU properties	135
57.	Configuring remote LU properties: general page	136
58.	Configuring remote LU properties: options page	137
59.	Selecting a CPIC symbolic name	138
60.	Configuring the CPIC properties: general page for HOSTR	139
61.	Configuring CPIC properties: partner information page for HOSTR	140
62.	Configuring CPIC properties: general page for HOSTS	141
63.	Configuring CPIC properties: partner information for HOSTS	142
64.	Configuring the local transaction program for the Send function	143
65.	Configuring the local transaction program for the Receive function	144
66.	Communications Server SNA Node configuration window	146
67.	Define the node window: basic page	147
68.	Define the node window: advanced page	148
69.	Define the node window: DLU requester page	149
70.	Define a LAN device window: basic page	150
71.	Define a LAN device window: advanced page	151
72.	Define a LAN device window: performance page	152
73.	Define a LAN connection window: basic page	153
74.	Define a LAN connection window: advanced page	154
75.	Define a LAN connection window: security page	155
76.	Define a partner LU 6.2 window: basic page	156
77.	Define a partner LU 6.2 window: advanced page	156
78.	Define a mode window: basic page	157
79.	Define a mode window: advanced page	158
80.	Define a local LU 6.2 window	159
81.	Define CPI-C side information basic window for the receive function	160
82.	Define CPI-C side information window: security page	161
83.	Define CPI-C side information basic window for the send function	162
84.	Define CPI-C side information window: security page	163
85.	Define a transaction program basic window for the send function	164
86.	Define a transaction program window: advanced page	165
87.	Define CPI-C side Information basic window for the receive function	166
88.	Define a transaction program window: advanced page	167
89.	Open window	168
90.	Node characteristics window	169
91.	Node characteristics connection selection window	170
92.	Node characteristics connection window	170
93.	Personal Communications SNA node configuration window	174

94.	Define the node: basic page	175
95.	Define the node: advanced page	176
96.	Define the node: DLU requester page	177
97.	Define a LAN device: basic page	178
98.	Define a LAN device: activation page	179
99.	Define a LAN device: performance page	180
100.	Define a LAN connection: basic page	181
101.	Define a LAN connection: advanced page	182
102.	Define a LAN connection: security page	183
103.	Define a partner LU 6.2: basic page	184
104.	Define a partner LU 6.2: advanced page	184
105.	Define a mode: basic page	185
106.	Define a mode: advanced page	186
107.	Define a local LU 6.2	187
108.	Define CPI-C side information: basic page, for the RECEIVE function	188
109.	Define CPI-C side information: security page	189
110.	Define CPI-C side information: basic page, for the SEND function	190
111.	Define CPI-C side information: security page	191
112.	Configuring the local Transaction Program, for the SEND function	192
113.	Configuring the local Transaction Program, for the RECEIVE function	193
114.	Open	194
115.	Node characteristics	194
116.	Node characteristics connections selection	195
117.	Node characteristics connections	195
118.	Software distribution environment	201
119.	TME 10 Software Distribution common - icon view folder	203
120.	Software Object profiles window	203
121.	New software profile window	204
122.	MY.Sample - basic configuration general Page	205
123.	MY Sample - basic configuration files Page	206
124.	Open window	206
125.	MY Sample - basic configuration files page	207
126.	File options window	208
127.	New software profile window	209
128.	Software Object profiles progress indicator	209
129.	Software Object profiles informational message	210
130.	Software Objects profile Window	210
131.	TME 10 Software Distribution common - icon view folder	211
132.	Software Object profiles window	212
133.	New software profile window	212
134.	My.Sample.Remote - advanced configuration general page	213
135.	My.Sample.Remote - advanced configuration remote directory page	214
136.	New software profile window	216
137.	Software Object profiles progress indicator	216
138.	Software Object profiles informational message	217
139.	Software Objects profile window	217
140.	My.Sample.Dynamic - advanced configuration general page	218
141.	My.Sample.Dynamic - advanced configuration files page	219

142.	Dynamic section - create notebook - general page	220
143.	Dynamic section - create notebook - condition page	221
144.	Dynamic section - create notebook - general page	222
145.	Dynamic section - create notebook - condition page	223
146.	MY Sample.Dynamic - advanced configuration files page	224
147.	New software profile window	225
148.	Software Object profiles progress indicator	225
149.	Software Object profiles informational message	226
150.	TME 10 Software Distribution - icon view folder	228
151.	Software Object profiles window	228
152.	Create a Software Object profile using DiskCamera window	229
153.	Take a picture of a disk drive window	229
154.	Copying files to the DiskCamera drive	230
155.	Editing the system environment variables	231
156.	Software Object profile generator window	232
157.	Informational message	233
158.	Create a Software Object profile using DiskCamera window	234
159.	Software Object profiles window with XLATOR added	234
160.	XLATOR - advanced configuration notebook	235
161.	XLATOR - advanced configuration notebook showing the files page	236
162.	XLATOR - advanced configuration notebook showing the variables page	237
163.	Software Object profile window	238
164.	Software Object profiles progress indicator	238
165.	Software Object profiles information message	239
166.	Discovered software package	246
167.	Example of a check script used for quiesce checks	250

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	76
13.	Target Definition Parameters	89
14.	STS Configuration File Parameters	98
15.	STS Connection Configuration File Parameters	99
16.	Target Identification Terms	101
17.	SNA/DS Configuration File Parameters	106
18.	SNA/DS Connection Configuration File Parameters	107
19.	Target Identification Terms	112

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 Windows NT, 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 Windows NT 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 Windows NT server.
- Chapter 9, “Installing and Configuring the TME 10 Software Distribution Server for Windows NT or 2000” explains how to install a TME 10 Software Distribution server and shows how to customize the installation for NetBIOS and IPX.
- Chapter 10, “Editing the Base Configuration File” explains how to edit the base configuration file.
- Chapter 11, “Defining Users and Targets” explains how to define users and targets.
- Chapter 12, “Configuring STS Remote Connection Files” describes how to configure remote connections in STS networks.
- Chapter 13, “Configuring SNA/DS Remote Connection Files” describes how to configure remote connections in SNA/DS networks.
- Chapter 14, “Configuring Microsoft SNA Server” describes how to configure the Microsoft SNA Server.
- Chapter 15, “Configuring IBM Communications Server” describes how to configure IBM Communication Server.
- Chapter 16, “Configuring IBM Personal Communications” describes how to configure IBM Personal Communication.

Part 3, “TME 10 Software Distribution Scenarios”

This part of the book includes the following chapter:

- Chapter 17, “Software Distribution Scenarios” presents typical user scenarios for preparing software for distribution and using DiskCamera.

Part 4, “Appendixes”

The manual has the following appendixes:

- Appendix A, “Implementing Inventory Discovery” provides a description of the inventory discovery procedure.
- Appendix B, “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 C, “Writing Change Control Scripts” describes change management scripts and how to create them.
- Appendix D, “Writing User Exits” on page 255 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 E, “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.

Table 1. TME 10 Software Distribution, Version 3.1.5 Platform Support

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.

<i>Table 2. Products from which TME 10 Software Distribution, Version 3.1.5 can be upgraded</i>		
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 MVW 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*

Changes to Statuses Reported by 'stattg'

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 OS/2) 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*.

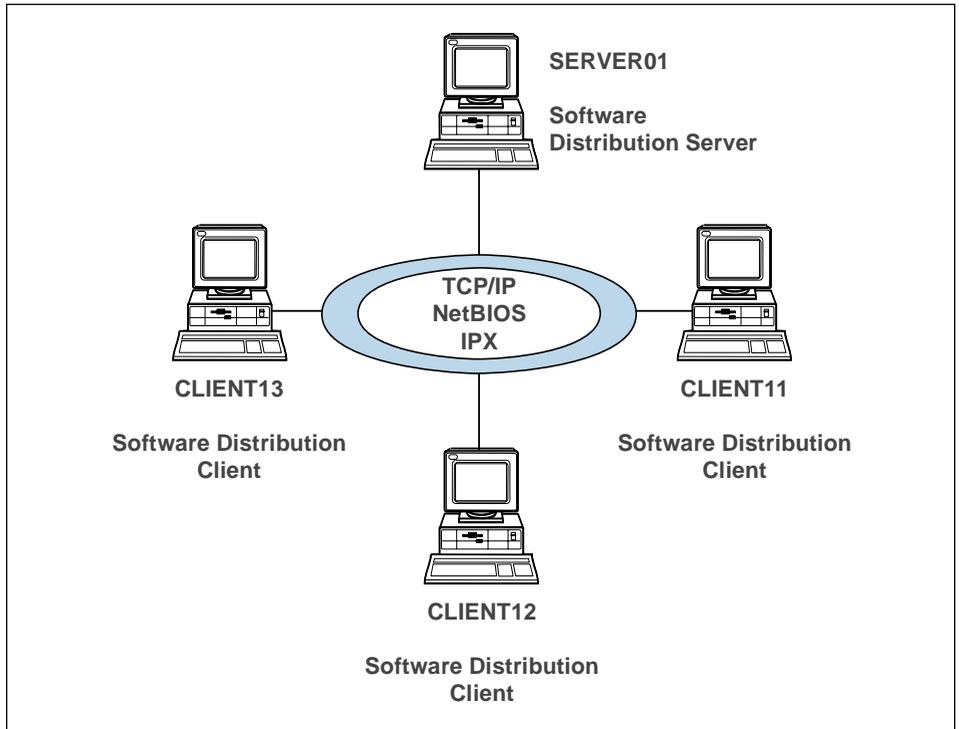


Figure 1. A Single-server TME 10 Software Distribution network

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

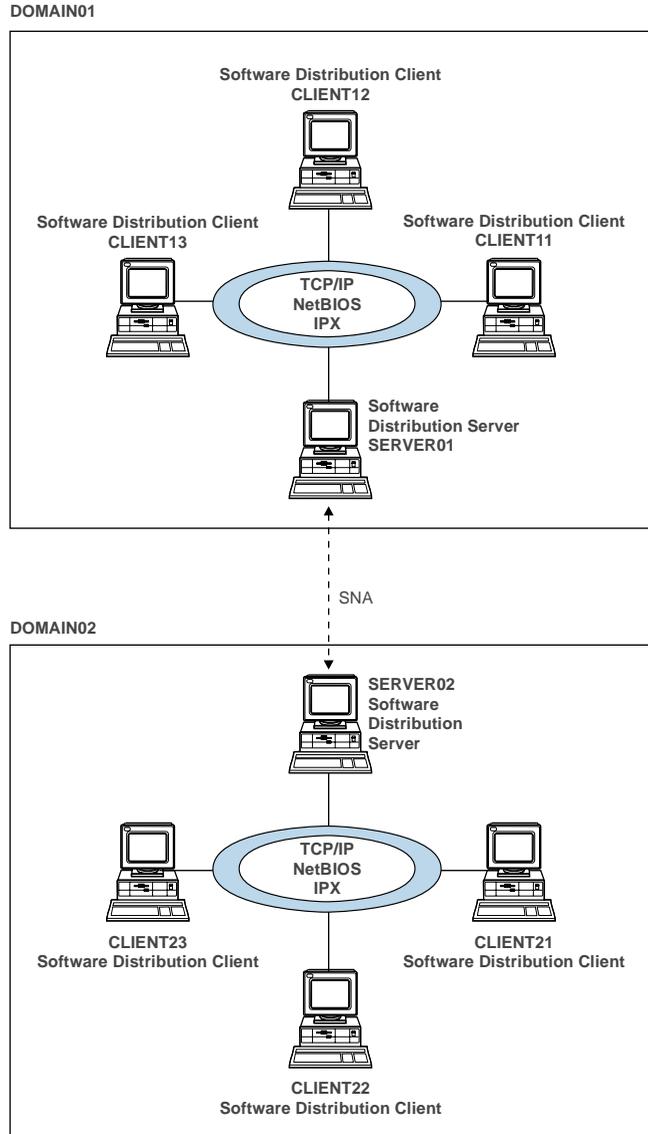


Figure 2. Interconnected domains

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.

Using TME 10 Software Distribution in a Network

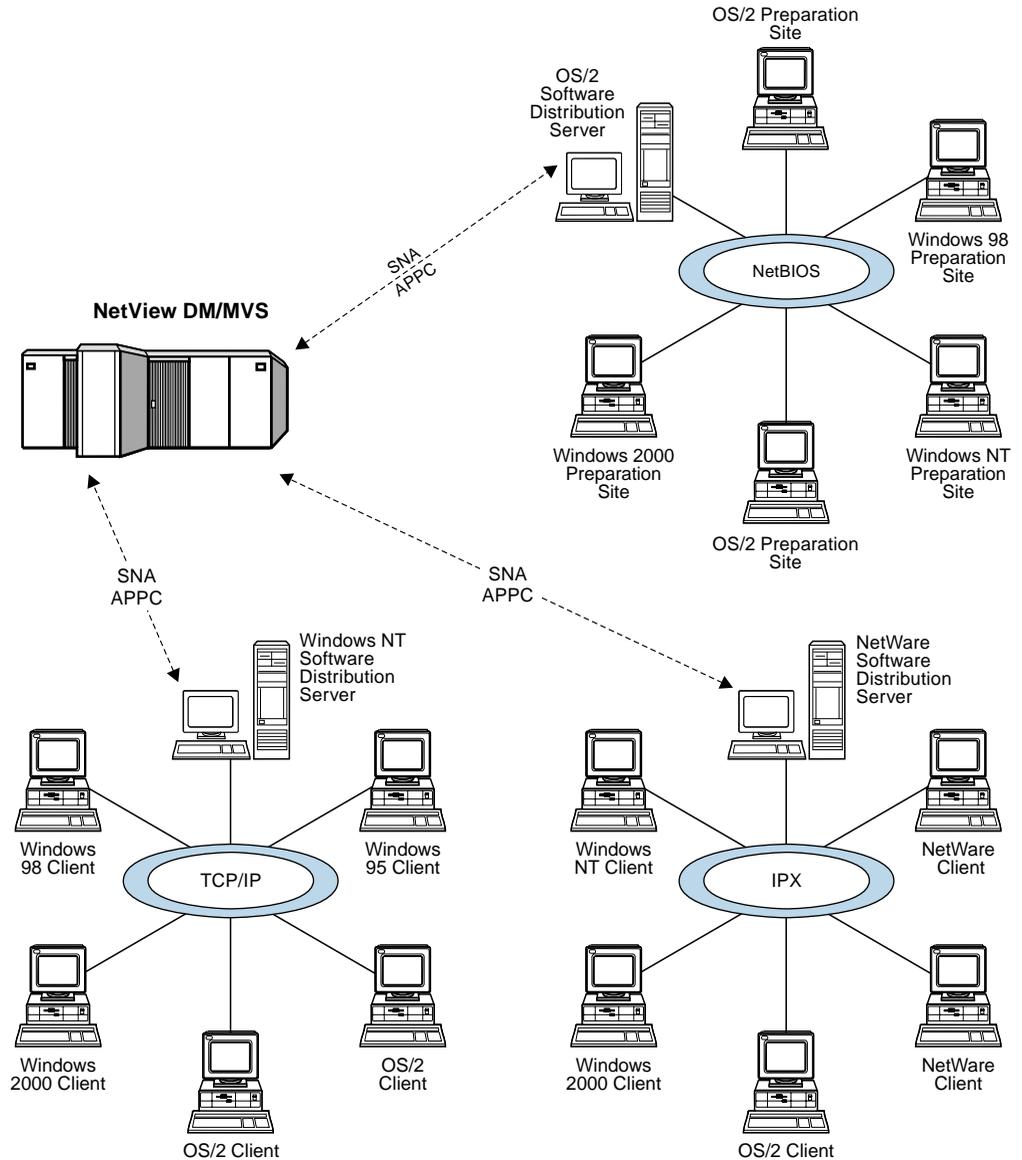


Figure 3. Software distribution from MVS

Using TME 10 Software Distribution in a Network

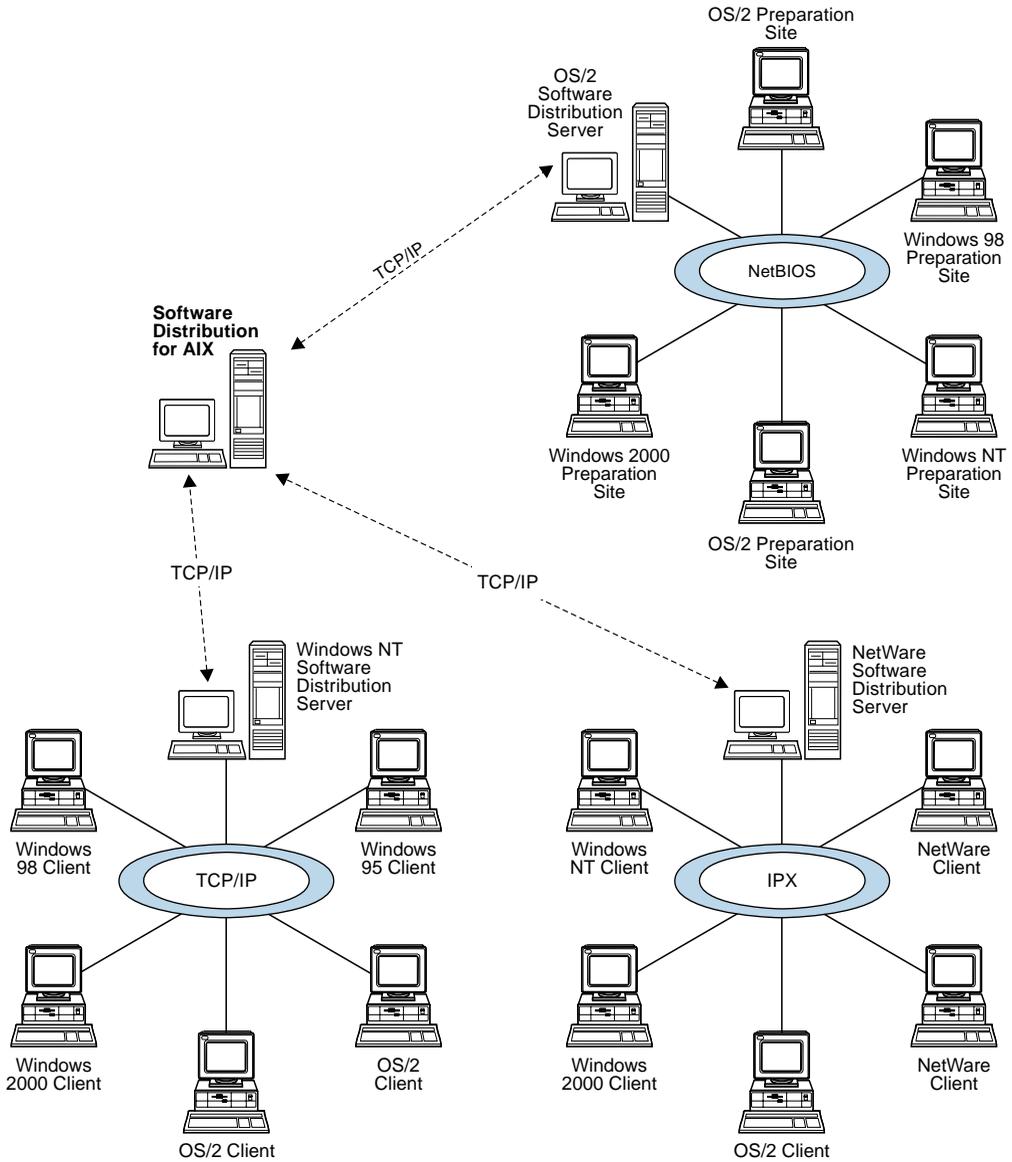


Figure 4. Software distribution from AIX

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

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.

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

TME 10 Software Distribution User Interfaces

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

Software Preparation Interface

The first window that appears in this interface 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.

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 `nvdml`. 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 Software Preparation icon. The TME 10 Software Distribution folder is displayed, as shown in Figure 5.

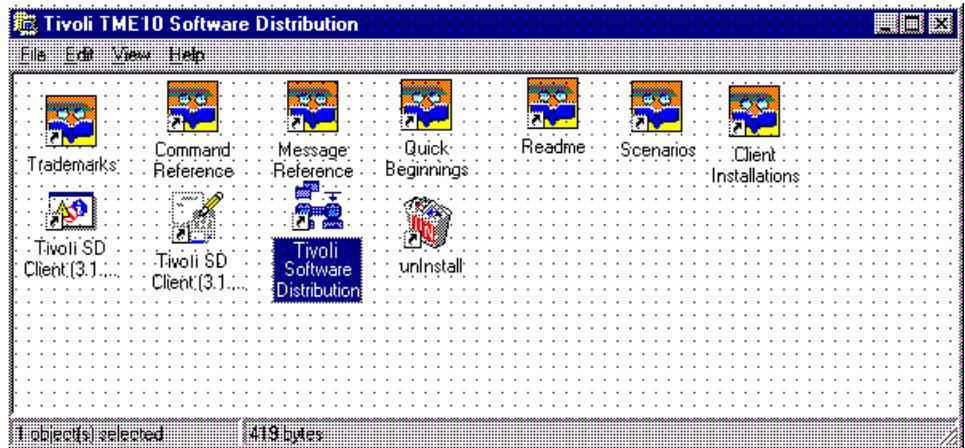


Figure 5. TME 10 Software Distribution for Windows NT folder

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:

Starting TME 10 Software Distribution

`nvdms 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 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: `nvdmgilg`
 - To start the software preparation interface, enter: `nvdmgipr`

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.

The screenshot shows a dialog box titled "Tivoli Software Distribution for Windows NT Logon". At the top, it states "The password will not be displayed". Below this are three input fields: "User ID", "Password", and "Server name". The "Server name" field contains the text "It0037a0". Below the input fields is a list box labeled "Servers" which contains the entry "It0037a0". At the bottom of the dialog are three buttons: "Logon", "Cancel", and "Help".

Figure 6. Logging on to TME 10 Software Distribution

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

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:

```
nvdms 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 **updcm** 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 75 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, or IPX

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

Grouping Targets to Facilitate Change Control Operations

“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 Windows NT 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.

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, REMREPNTOP). 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 `fn dinv`, and it creates at each target output files (`fn dswinv`, `fn dhwinv`, and `fn dtkinv`) 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 **`nvd m inv`** command.
- From the graphical interface, select **Inventory** from the **Selected** pull-down menu in the Targets window.

Note that for `fn dinv` 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.

Tracking the Progress of Change Control Operations

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

Table 7. Administrator Profile (FNDADMN)

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.

Table 8. Builder Profile (FNDBLD)

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 `nvdml` 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.

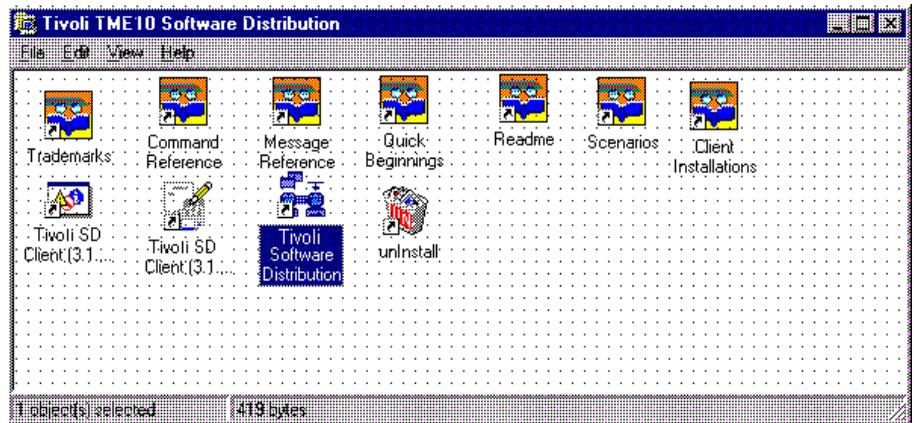


Figure 7. Distribution Server documentation - icon view

Viewing Online Help

The icons in the folder represent these items:

- Quick Beginnings, an INF file (FNDNTMST.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 Windows NT 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 Windows NT 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 Windows NT.
- 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.
- 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 XVIEW 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 Windows NT 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:

```
XVIEW 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 MS-DOS 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:

```
XVIEW 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 Windows NT, 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 and Configuring the TME 10 Software Distribution Server for Windows NT or 2000” on page 53
- Chapter 10, “Editing the Base Configuration File” on page 75
- Chapter 11, “Defining Users and Targets” on page 89
- Chapter 12, “Configuring STS Remote Connection Files” on page 97
- Chapter 13, “Configuring SNA/DS Remote Connection Files” on page 105
- Chapter 14, “Configuring Microsoft SNA Server” on page 117
- Chapter 15, “Configuring IBM Communications Server” on page 145
- Chapter 16, “Configuring IBM Personal Communications” on page 173

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 a TME 10 Software Distribution Server.

You must be logged on with administrator authority to install and configure a TME 10 Software Distribution server for Windows NT or Windows 2000.

Before you install TME 10 Software Distribution, Version 3.1.5 for Windows NT, ensure that you have the required hardware and software installed on your workstations.

The hardware and software required for TME 10 Software Distribution depend on how you choose to set up the network environment and the number of clients to be managed.

Supported network environments are:

- Single LAN with a single TME 10 Software Distribution server
- Multiple LANs, each with its own TME 10 Software Distribution server

A TME 10 Software Distribution server is necessary for using TME 10 Software Distribution. If you choose to set up multiple logical work groups, multiple TME 10 Software Distribution servers are required, one for each logical work group. Note that multiple TME 10 Software Distribution servers can communicate with one another.

The number of TME 10 Software Distribution servers that are required in a specific environment depends on the number of active clients, the use of multiple communication protocols, and the existence of logically separate work groups.

TME 10 Software Distribution Server Prerequisites

These are the minimum requirements for a LAN configuration that includes a single TME 10 Software Distribution server.



These prerequisites apply to installation only. The requirements will increase when many clients are active.

Hardware

- 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 35 MB
- Enough disk space to store the software objects for distribution.
- Token-ring or Ethernet card

Hardware and Software Prerequisites

- CD-ROM drive for product installation

Software

The following is required on the TME 10 Software Distribution server workstation:

- One of the following:
 - Windows NT 3.51
 - Windows NT 4.0 plus Service Pack 5 to 6a
 - Windows 2000 Professional
 - Windows 2000 Server
- If you need a connection for NetView DM for MVS, one of the following is required:
 - Microsoft SNA Server 4.0 or later for Windows NT
 - Microsoft SNA Server 4.0 with SP3 or later for Windows 2000
 - Personal Communications 4.1 with APAR ICI8696
 - Communications Server 5.0.2 or later for Windows NT
 - Communications Server 6.1 or later for Windows 2000

Communication Protocols

Server and client communication supports NetBIOS, TCP/IP, and IPX/SPX. These communication protocols are embedded and supported in all Windows Operating Systems.

A routing table is required for communication between servers.

Protocols to Connect to Another Server

The following protocols are supported for a Windows NT or Windows 2000 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 Potocol
OS/2	IPX NetBIOS TCP/IP
NetWare	IPX TCP/IP
Windows NT or Windows 2000	IPX NetBIOS TCP/IP
AIX	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 a Windows NT or Windows 2000 server communicating with a TME 10 Software Distribution Client.

Table 11. Protocols to Connect to A Client

Platform	Communications Potocol
OS/2	IPX NetBIOS TCP/IP
NetWare	IPX TCP/IP
Windows NT or Windows 2000	IPX NetBIOS TCP/IP
Windows 95 or Windows 98	IPX NetBIOS TCP/IP
Windows 3.11	IPX 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.

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 and Configuring the TME 10 Software Distribution Server for Windows NT or 2000

This chapter explains how to install and configure a TME 10 Software Distribution Server. It includes a description of:

- How to install the product (“Installing TME 10 Software Distribution for Windows NT or Windows 2000 in Attended Mode”)
- How to configure for NetBIOS (“Customizing the Server for NetBIOS” on page 61)
- How to configure for IPX (“Customizing the Server for IPX” on page 64)
- How to reinstall the product in attended mode (“Upgrading TME 10 Software Distribution for Windows NT” on page 66)
- How to use a response file to install the product (“Using a Response File to Install a TME 10 Software Distribution Server” on page 70)
- How to uninstall the product (“Uninstalling a TME 10 Software Distribution Server or Windows NT or Windows 2000” on page 70)

Installing TME 10 Software Distribution for Windows NT or Windows 2000 in Attended Mode

This section describes how to install TME 10 Software Distribution for Windows NT or Windows 2000 using the TCP/IP communication protocol. See “Changing Protocols after the Installation” on page 61 for information on how to customize an installation for NetBIOS and IPX.

If your video display is set to monochrome, you will not be able to distinguish in the list of components the components that you select to install. If you must use a monochrome video, look at the **Space required** field in the Select Components window to see which components you selected for the installation.

To install the TME 10 Software Distribution server, complete the following steps:

- 1 Insert the TME 10 Software Distribution CD-ROM (number LK3T-5087-00)
- 2 Type the following command from the SD4WNT9x directory on the CD-ROM:
setup

The Welcome window appears.

Installing a TME 10 Software Distribution Server in Attended Mode

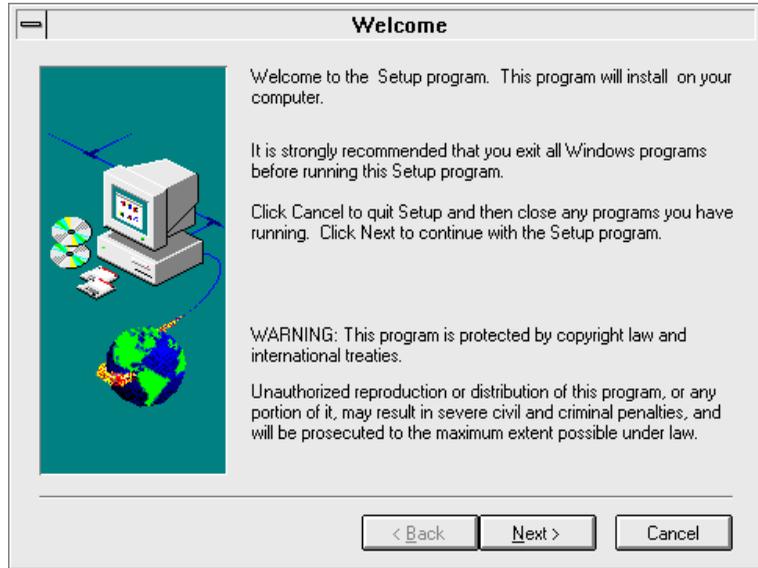


Figure 8. Windows NT or Windows 2000 server - installation welcome window

3 Select **Next** to continue with the installation.

The Select Installation window appears.

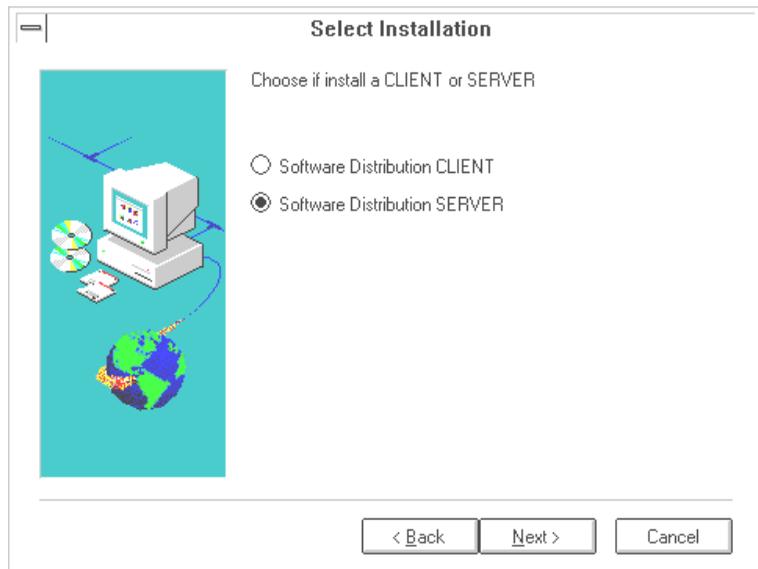


Figure 9. Windows NT or Windows 2000 server - select installation window

On this window you specify whether to install the software distribution server or the client.

Installing a TME 10 Software Distribution Server in Attended Mode

4 Select **Software Distribution SERVER**, and then select **Next**.

The Select Components window appears.

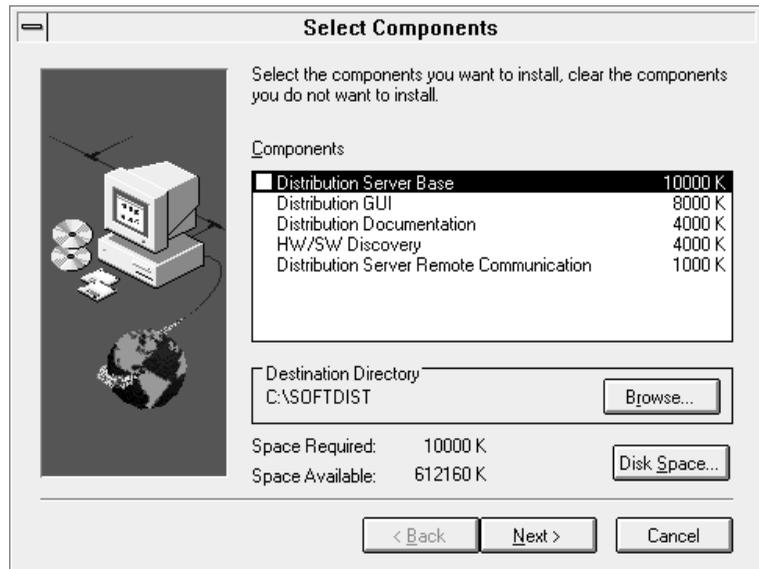


Figure 10. Windows NT or Windows 2000 server - select components window

5 Select the components you want to install from the list:

- The **Distribution Server Base** component is mandatory.
- The **Distribution GUI** component contains the user interface. It is optional.
- The **Distribution Documentation** component contains the product documentation. It is optional.
- **HW/SW Discovery** component. It is optional.

The hardware/software discovery tool contains 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.

- The **Distribution Server Remote Communication** component customizes the `nvdn.cfg` file.

It is optional. If you install this component, the server can communicate both with other servers and with the clients. If you do not install this component, the server can communicate with the clients only.

On the Select Components window you can also select **Browse** and **Disk Space**.

If you select **Browse**, the Choose Directory window appears. You can use this window to change the destination directory.

Installing a TME 10 Software Distribution Server in Attended Mode

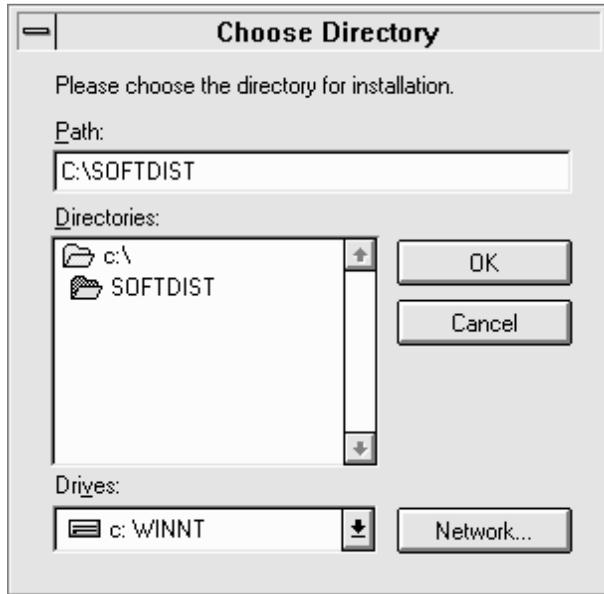


Figure 11. Choose directory window

If you select **Disk Space**, the Available Disk Space window appears.



Figure 12. Available disk space window

You can use this window to see the amount of disk space that is available. Select a drive, and select **OK**.

The Select Components window reappears.

- 6 After you make your choices on the Select Components window, select **Next**. The Select Program Folder window appears.

Installing a TME 10 Software Distribution Server in Attended Mode

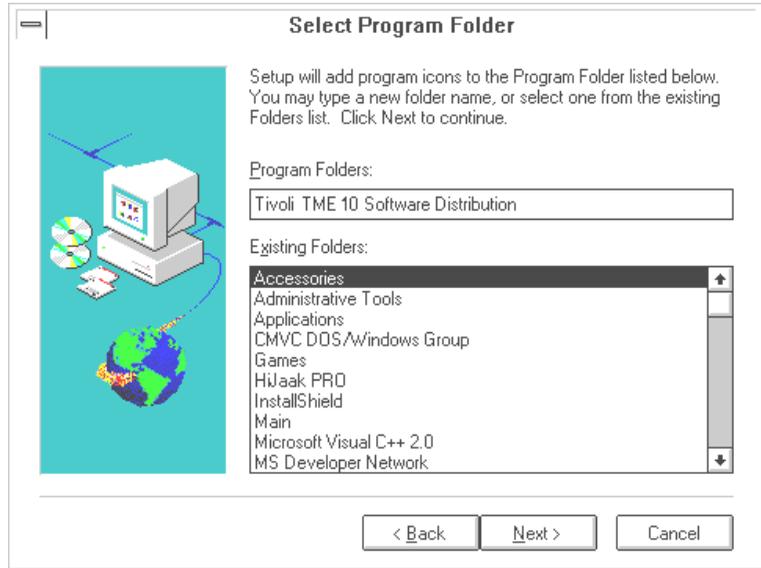


Figure 13. Windows NT or Windows 2000 server - select program folder window

- 7 After you make your choices on the Select Program Folder window, select **Next**. The Select Protocol window appears.

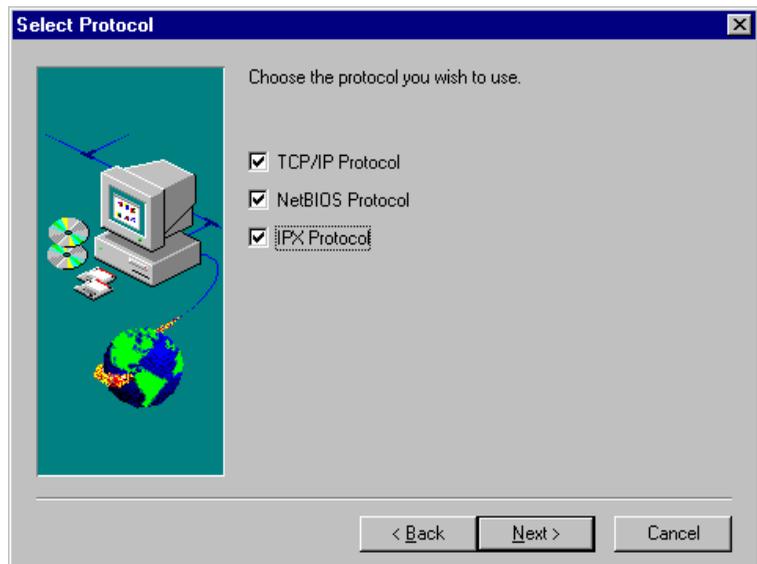


Figure 14. Windows NT or Windows 2000 server - select protocol window

Choose the connection protocol you need, you can select one, two, or all three protocols, and then select **next**.

Installing a TME 10 Software Distribution Server in Attended Mode

The Server Configuration window appears.

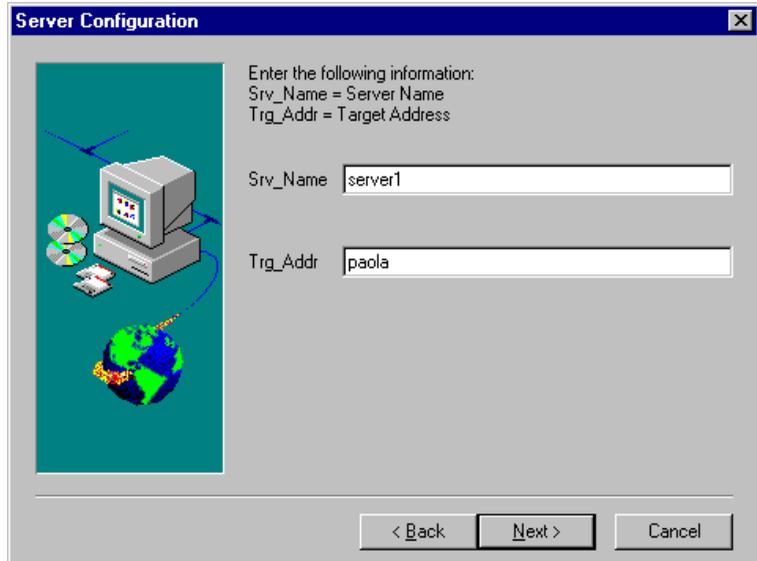


Figure 15. Windows NT or Windows 2000 server - server configuration window

8 Enter the **server name** and the **target address**, and select **Next**.

The Server Configuration window changes to allow you to enter the server name of the connection protocol you selected before, then select **Next**. If you selected IPX remember that the server IPX name is composed of the following:

- (NetId) Is the network ID.
- (AdAdd) Is the adapter address.
- (AplAdd) Is the application address. It must be 869F.

Installing a TME 10 Software Distribution Server in Attended Mode

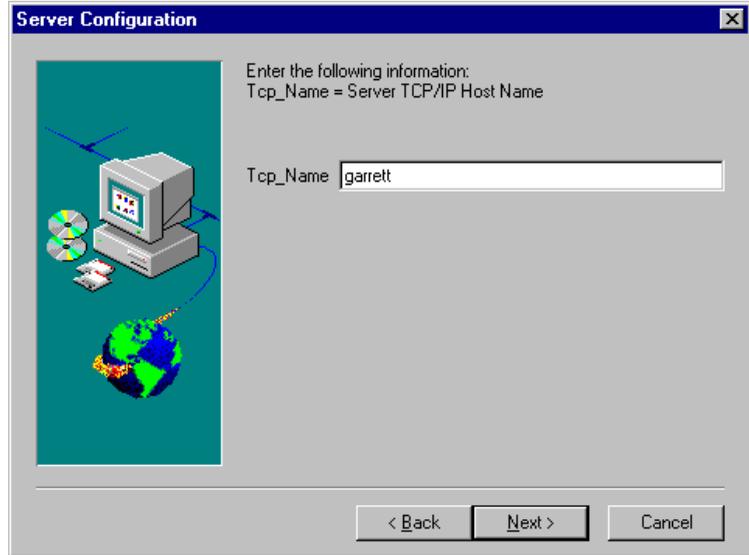


Figure 16. Windows NT or Windows 2000 server - ServerName configuration window

The User Name Configuration window appears.

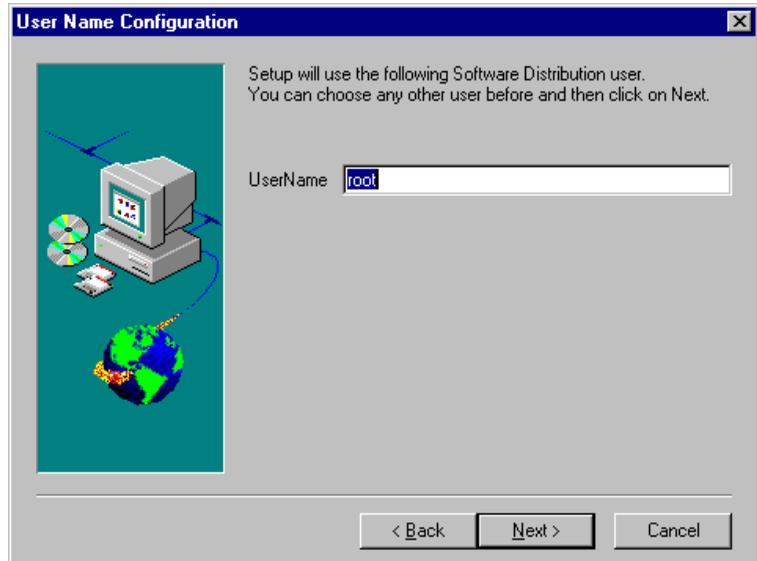


Figure 17. Windows NT or Windows 2000 server - user name configuration window

Enter the **user name**, and select **Next**.

The Start Product Selection window appears.

Installing a TME 10 Software Distribution Server in Attended Mode



Figure 18. Windows NT or Windows 2000 server - start product selection window

9 Select either **Automatic Startup** or **Manual Startup**.

If you choose **Automatic Startup**, the program defines a service from the Windows NT or Windows 2000 point of view, and the product will start automatically.

10 After you make your choice, select **Next**.

The Copying Profiles window appears. This window displays the progress of the installation.

When the installation is complete, the Setup Complete window appears.



Figure 19. Windows NT or Windows 2000 server - setup complete window

- 11 Specify whether or not you want to restart your computer now, and select **Finish**. The installation process for the server is complete.

Changing Protocols after the Installation

This section describes how to customize a TME 10 Software Distribution for Windows NT or Windows 2000 installation for the NetBIOS and IPX communication protocols on the server and client.

Customizing the Server for NetBIOS

If you have already performed the server installation using the NetBIOS protocol skip the following step and go to step 4. If you want to change the installation protocol to use NetBIOS complete the following steps:

- 1 In the NVDM.CFG file, in the main installation directory, remove the comment from the row that contains

```
#PROTOCOL: NBI (machine_name) 0 10
```

To remove the comment, delete the # from the first column.

- 2 Comment out the row that contains

```
PROTOCOL: TCP (machine_host_name) 729 50
```

To comment something out, insert a # in the first column.

- 3 Save the file.
- 4 From the Program Manager, select the main menu window.

Customizing for NetBIOS

5 Select Control Panel

The Network Setting window appears.

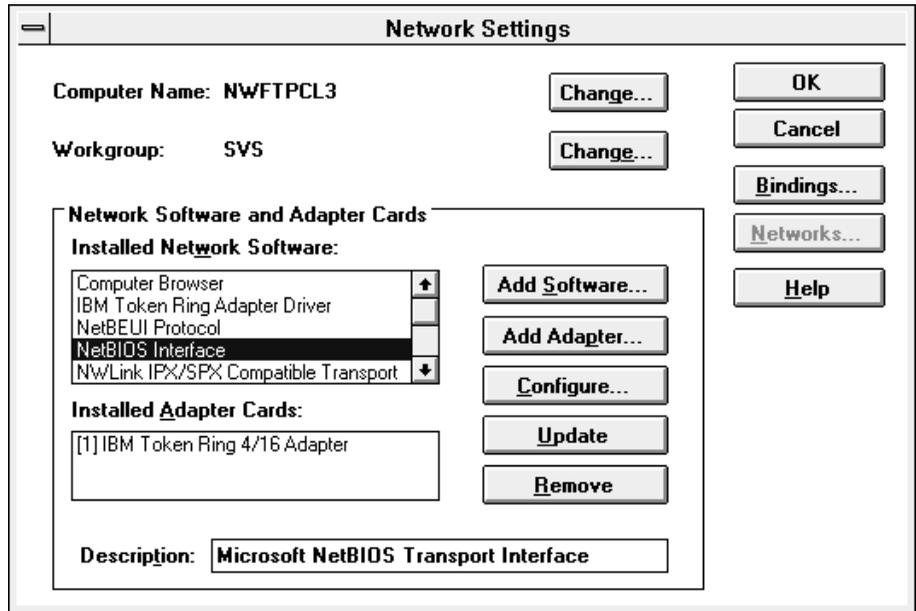


Figure 20. Windows NT or Windows 2000 network settings window

6 Select NetBIOS interface from the list of network settings and then Select Configure....

The NetBIOS Interface window appears.

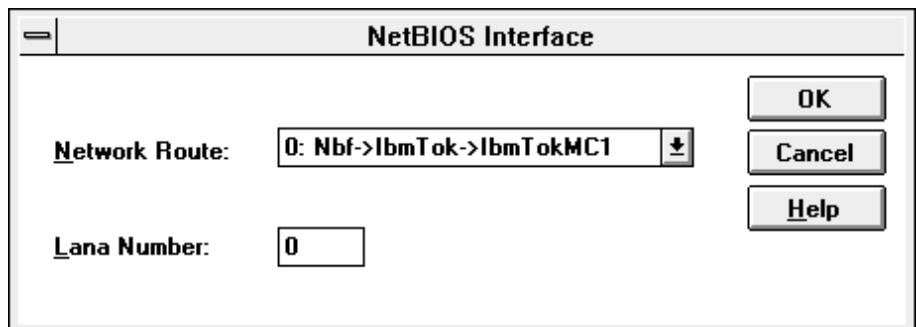


Figure 21. NetBIOS interface window

7 In the Network route field, select network route.

The NetBIOS Interface window changes, as follows:

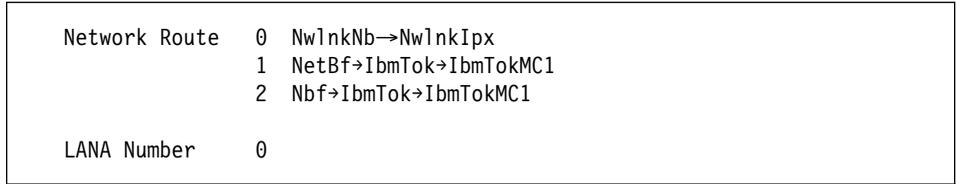


Figure 22. Example of the NetBIOS interface window

- 8** In the Network route field, select network route **Nbf→IbmTok→IbmTokMC1**.
The window changes, as in the following example:

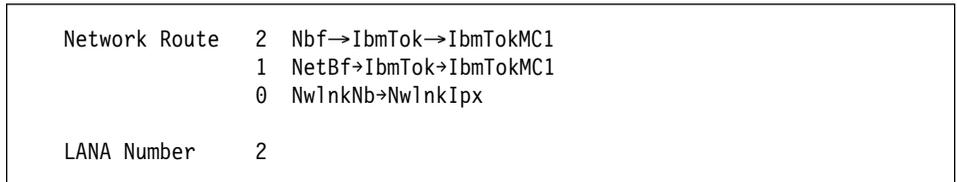


Figure 23. Example of the NetBIOS interface window

- 9** Change the LAN adapter number from **2** to **0** by clicking on the scroll bar on the LAN Adapter Number field until a **0** appears. The window changes, as in the following example:

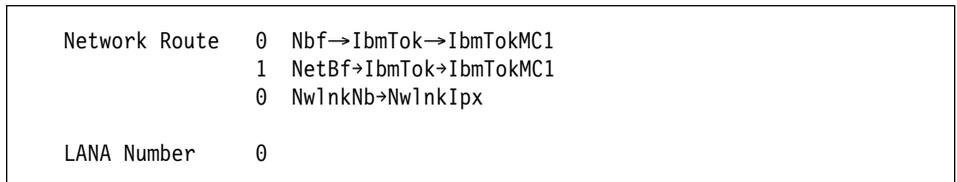


Figure 24. Third example of the NetBIOS interface window

- 10** Change the LAN adapter number of network route **NwlnkNb→NwlnkIpx** from **2** to **0**:
- Select **NwlnkNb→NwlnkIpx**.

Customizing for IPX

- Change the LAN adapter number from **0** to **2**. The window changes, as in the following example:

Network Route	0	Nbf→IbmTok→IbmTokMC1
	1	NetBf→IbmTok→IbmTokMC1
	2	Nw1nkNb→Nw1nkIpx
LANA Number	2	

Figure 25. Fourth example of the NetBIOS Interface window

This is the procedure to follow when you have only one LAN adapter card. If you have more than one, you must:

- Configure the NetBIOS Interface window for each card.
- Add the correct LAN adapter number to the PROTOCOL keyword in the NVDM.CFG. file:

```
PROTOCOL: NBI (machine_name) (Lanadapternumber) 10
```

11 Save the file.

The NetBIOS customization procedure for the server is now finished.

Customizing the Client for NetBIOS

If you have already performed the client installation using the NetBIOS protocol skip this section. If you want to change the client installation protocol to use NetBIOS perform the following additional steps:

1 In the NVDM.CFG file, comment out the rows that contain:

```
SERVER: (server_name) TCP (server_name)
```

```
PROTOCOL: TCP (client_host_name) 729 50
```

To comment something out, insert a # in the first column.

2 Remove the comment from the rows that contain:

```
#SERVER: (server_name) NBI (server_name)
```

```
#PROTOCOL: NBI (client_NBI_name) 0 50
```

To remove the comment, delete the # from the first column.

3 Save the file.

Customizing the Server for IPX

If you have already performed the server installation using the IPX protocol skip this section. If you want to change the installation protocol to use IPX, complete the following steps:

- 1 In the NVDM.CFG file, in the main installation directory at the client site, remove the comment from the row that contains:

```
#PROTOCOL: IPX (cNetId) (cAdpAdd) (cApIAdd) 50
```

To remove the comment, delete the # from the first column.

- 2** At the server site, remove the comment from the row that contains:

```
#PROTOCOL: IPX (sNetId) (sAdpAdd) (sApIAdd) 50
```

- 3** Comment out the row that contains:

```
PROTOCOL: TCP (machine_name) 729 50
```

To comment something out, insert a # in the first column.

- 4** Enter the correct values for the following:

(NetId) Is the network ID.

(AdAdd) Is the adapter address.

(ApIAdd) Is the application address. It must be 869F.

- 5** Save the file.

The IPX customization procedure for the server is now finished.

Customizing the Client for IPX

If you have already performed the client installation using the IPX protocol skip this section. If you want to change the client installation protocol to use IPX, perform the following additional steps:

- 1** In the NVDM.CFG file at the client site, comment out the rows that contain:

```
SERVER: (server_name) TCP (server_name)
```

```
PROTOCOL: TCP (client_host_name) 729 50
```

To comment something out, insert a # in the first column.

- 2** In the NVDM.CFG file at the client site, remove the comment from the rows that contain:

```
#SERVER: (server_name) IPX (sNetId) (sAdpAdd) (sApIAdd)
```

```
#PROTOCOL: IPX (sNetId) (sAdpAdd) (sApIAdd) 50
```

To remove the comment, delete the # from the first column.

- 3** Enter the correct values for the following:

(NetId) Is the network ID.

(AdAdd) Is the adapter address.

(ApIAdd) Is the application address. It must be 869F.

- 4** Save the file.

Upgrading TME 10 Software Distribution for Windows NT

This section describes how to upgrade TME 10 Software Distribution for Windows NT to the latest release. After you have upgraded an existing installation to version 3.1.5 using this scenario, or have installed a version of TME 10 Software Distribution for Windows NT or Windows 2000 from scratch using one of the other scenarios, you can use these instructions to refresh the installation, in the event, for example, that you wish to install a component not installed originally. To upgrade or refresh a TME 10 Software Distribution server, perform the following steps:

- 1 Insert CD-ROM number *LK3T-5087-00*
- 2 Type the following command from the SD4WNT9x directory on the CD-ROM:

SETUP

The Welcome window appears.

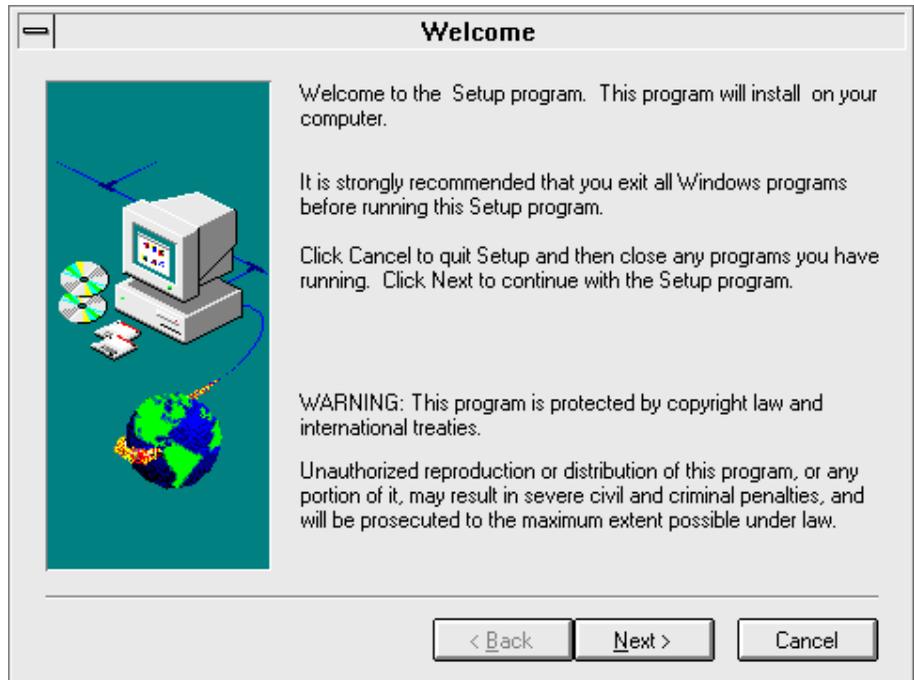


Figure 26. Windows NT server - installation welcome window

3 Select **Next** to continue with the installation.

The Select Client or Server window appears.

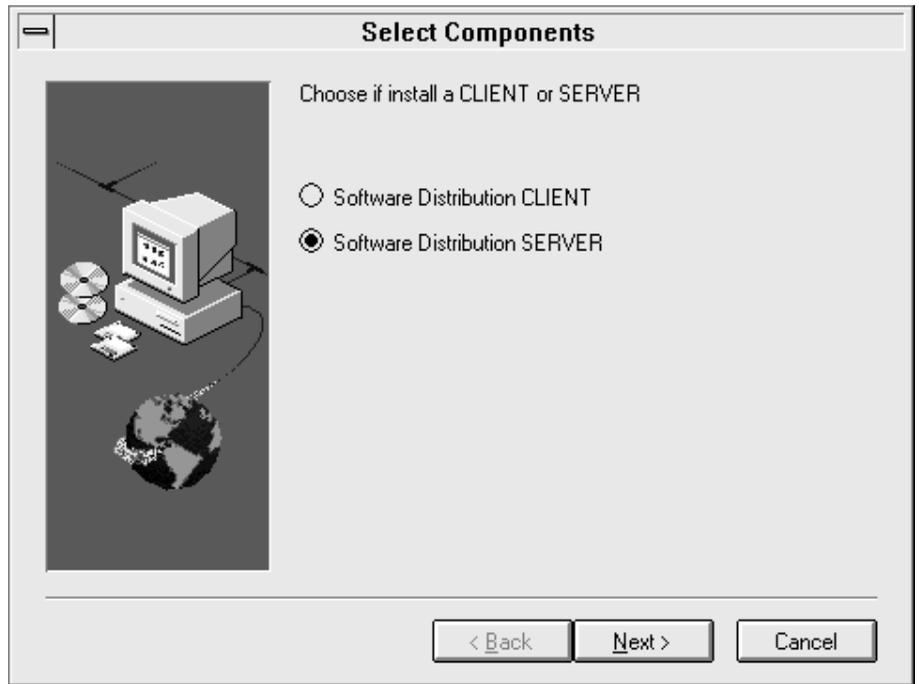


Figure 27. Windows NT server - select client or server window

On this window you specify whether to reinstall the software distribution server or the client.

4 Select **Software Distribution SERVER**, and then select **Next**.

The Reinstall Selection window appears.

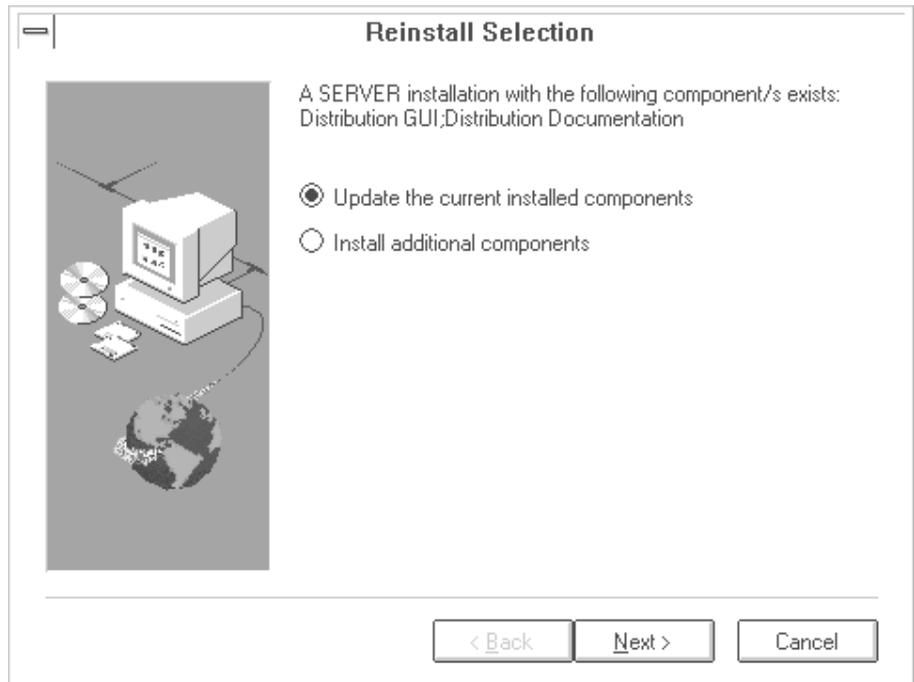


Figure 28. Windows NT server - reinstall selection window

On this window you specify whether you want to update the currently installed options or to install additional options. If you have previously installed all the options, only the first choice will be available:

- **Update the current installed components**

If you choose this option, the binary files are reloaded in the same directory as the old files, and the old ones are overwritten. The configuration files are not changed.

- **Install additional components**

If you choose this option, you will be able to install the components you did not previously install. The configuration files are not changed.

5 Specify one of the two options, and select **Next**.

If you selected Update the current installed components, the Copying Profiles window appears. This window displays the progress of the installation.

When the reinstallation is complete, the Setup Complete window appears. This is the same window that appears at the end of a first installation (see Figure 19 on page 61). Use this window to specify whether or not you want to restart your

computer now. After you choose, select **Finish**. The reinstallation process is complete.

If you selected Install additional components, the Select Components window appears.

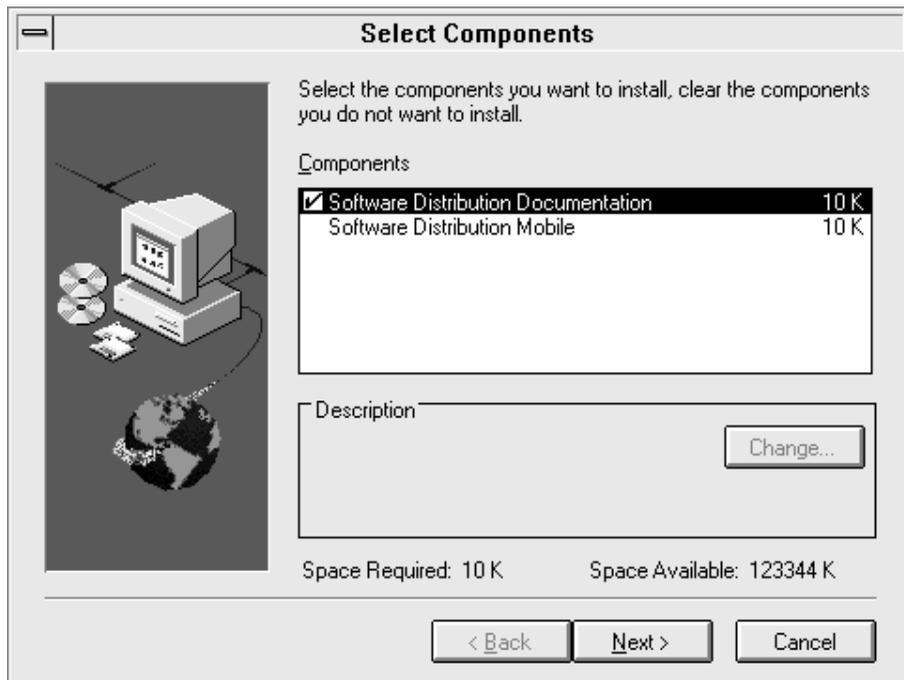


Figure 29. Windows NT server - select components window

- 6 Only the components that were not previously installed appear as choices. You cannot change the directory.

Specify the components you want to install, and select **Next**.

The Copying Profiles window appears. This window displays the progress of the installation.

When the reinstallation is complete, the Setup Complete window appears (see Figure 19 on page 61).

- 7 Specify whether or not you want to restart your computer now, and select **Finish**. The upgrade or refresh process for the server is complete.

Using a Response File to Install a TME 10 Software Distribution Server

You can install TME 10 Software Distribution for Windows NT or Windows 2000 in unattended, or silent, mode by using the sample response file that is provided with the TME 10 Software Distribution product on the CD-ROM. This sample file uses the TCP/IP communication protocol.

To install the TME 10 Software Distribution Server using a response file perform the following steps:

- 1 Copy the response file, SETUP.ISS, to a private directory.

- 2 Customize the file.

The file contains an explanation of the keywords and how to customize them. The lines of the file that you can customize are indicated with a string of asterisks. You must not modify lines that are not indicated with asterisks. The values you supply correspond to the values you supply on the installation panels during an attended installation.

- 3 Enter the following command:

```
SETUP -f1c:\<mydir>\setup.iss -s
```

You can check the results of the installation in the log file that is created by the -f2 option on the SETUP command, and in the file INSTLOG in the installation directory.

Uninstalling a TME 10 Software Distribution Server or Windows NT or Windows 2000

When you uninstall, you must delete the entire product. You cannot delete only a part of it.

Perform the following steps to uninstall the TME 10 Software Distribution server:

- 1 If the TME 10 Software Distribution product is running, stop it before you uninstall. To stop it, enter the following command:

```
NVDM STOP -x -k
```

- 2 Close the TME 10 Software Distribution GUI.

- 3 Select the **Uninstall** icon on the TME 10 Software Distribution group window.

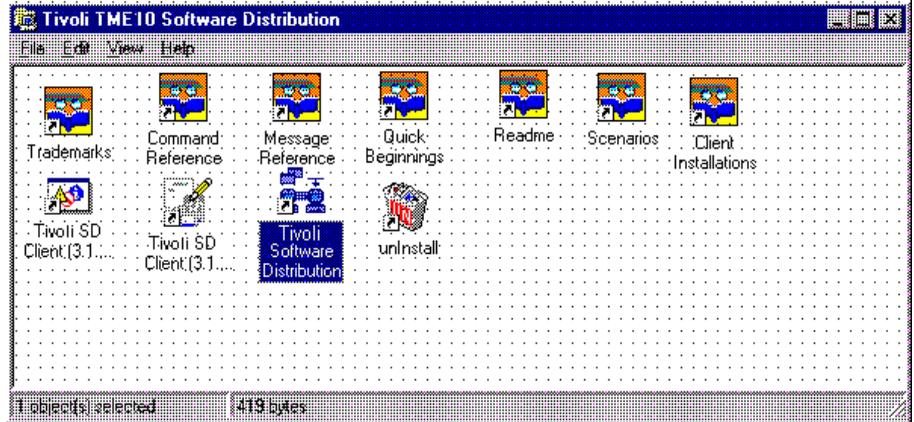


Figure 30. Windows NT or Windows 2000 server - uninstall icon

A panel that asks, “Are you sure that you want to completely remove selected application and all of its components?” appears.

4 Select YES

A panel named “Remove Programs from Your Computer” appears.

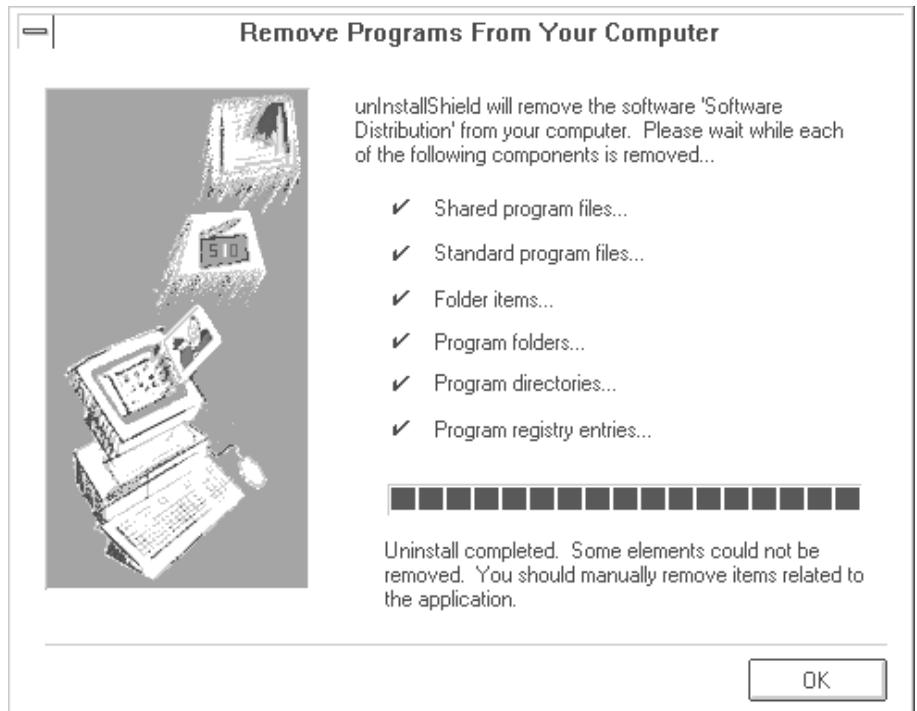


Figure 31. Windows NT or Windows 2000 server - uninstallation panel

Removing Settings if Uninstall Fails

5 Select **OK**.

A panel that says, "Please shut down your system before attempting a new installation of this product" appears.

6 Select **OK**.

Removing Settings if the Uninstall Process Fails

When you install TME 10 Software Distribution for Windows NT or Windows 2000, the product sets some variables for the system environment and the Registry keys. If the uninstall process fails, you must perform the following steps to remove these variables and keys:

If you are in a Windows NT 4.0 or Windows 2000 environment:

1. Select the **Start** push button then **Settings ▶ Control Panel**.
2. Select the System icon.
The System Properties window appears.
3. Select the Environment page.

If you are in a Windows NT Version 3.51 environment:

1. Select the Main icon from the Program manager window.
The Main window appears.
2. Select the Control Panel icon
The Control Panel window appears.

From any Windows NT or Windows 2000 environment

1. Delete the FND CFG, FNDUSER, and NVDMBASE variables from the System Variables scrolling list.
2. Select the Path variable from the System Variables scrolling list.
3. Delete the TME 10 Software Distribution path from the PATH variable, and then select **set**.
4. Select **OK**.
5. From a command prompt enter the following command:
`start regedt32`
The Registry Editor windows appears.
6. Select the HKEY_LOCAL_MACHINE key.
7. Select SYSTEM ▶ CurrentControlSet ▶ Control and delete the NetViewDM subkey.
8. Select SYSTEM ▶ CurrentControlSet ▶ Services and delete the NetViewDM and the NetViewDM(Act) subkeys.

9. Select SYSTEM ► CurrentControlSet ► Services ► SnaBase ► Parameters and delete the TPs key.
10. Repeat Steps 7, 8, and 9 for ControlSet00X, where X is a number from 1 to 3. The SYSTEM ► ControlSet00X are the backup subkeys for the CurrentControlSet subkeys.
11. Select SOFTWARE and delete the following keys:
 - SWOBJPRF.INI
 - NVDMGILG.INI
 - SWPRFINF.INI
12. Select the SOFTWARE ► IBM hive and delete the Software Distribution key.
13. Close the Registry Editor.
14. Reboot the workstation.

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 Windows NT system are at the TME 10 Software 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 TME 10 Software Distribution server configuration is performed during the installation process itself and stored in the base configuration file. This section describes the format of this file and the values that you can enter in its fields.

Base Configuration File Parameters

The base configuration file, <prod_>\nvdm.cfg, contains the system parameters for controlling TME 10 Software Distribution, Version 3.1.5 for Windows NT. The nvdm.cfg file generated at installation does not contain all possible parameters, only a default subset of them. You can add or remove parameters and their values by editing the file using any text editor.

The file is stored in a fixed text format. Each line starts with one of the keywords described in Table 12 on page 76. Keywords must be entered in uppercase and ended with a colon (:). Each keyword can be used only once with the exception of the **SERVER** keyword, which can be used up to five times. The order of keywords in the file is not important, and blank and comment lines can be included. Comment lines begin with a hash character (#).

To make modifications to the file operative, you must stop and then restart the product by using the commands:

```
nvdms stop -x -k  
nvdms start
```

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

Editing the Base Configuration File

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, Version 3.1.5 for Windows NT 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 uses more disk space. A small value degrades performance very slightly, because the trace is backed up more often.</p> <p>Refer to the online Command Reference for the product for information about the API trace file.</p>
AUTHORIZE	<p>This keyword authorizes a pull mode client to install a change file or to execute a data file.</p> <p>NONE No target is authorized to install a change file or to execute a data file unless it is authorized via the auth command.</p> <p>ALL Any target is authorized to install a change file or to execute a data file unless it is not authorized via the unauth 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 Windows NT. When you change the value of this keyword, all the authorized targets become unauthorized and vice versa.</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>

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

Keyword	Description
AUTOMATIC TARGET REGISTRATION	<p>This keyword enables clients to automatically configure themselves, or to autoregister, in the server database as one of the server's local targets. The configuration is performed the first time a client connects to the server, if the client is not already configured. The client is also registered at all servers at higher levels in the network hierarchy, if AUTOMATIC TARGET REGISTRATION is set to Yes in their base configuration files as well.</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>You must specify the TARGET ADDRESS and TARGET MODE keywords in the client's base configuration file if automatic registration is to be performed. In addition, set the CONFIGURATION keyword in the client base configuration file to CLIENT.</p> <p>When an automatic target registration is performed an inventory is automatically scheduled.</p>
BACKUP AREA	<p>The name of the directory that change files are stored in when they are installed as removable on a workstation. The default is <prod_>\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.</p>
CONFIGURATION	<p>The configuration of your TME 10 Software Distribution, Version 3.1.5 for Windows NT node. The name is set up during installation and cannot be changed. It can be one of the following:</p> <p>SERVER_NO_COMMS Base and server options installed.</p> <p>SERVER_WITH_COMMS Server and communication options installed.</p> <p>The value of the field in the TME 10 Software Distribution server base configuration file can be displayed using the lsbs command. (See <i>TME 10 Software Distribution, Version 3.1.5 for Windows NT Reference</i>.)</p>

Editing the Base Configuration File

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

Keyword	Description
CONNECTION WINDOW DURATION	<p>The value you specify on this parameter 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 DACA connection is considered to have failed.</p> <p>Enter a value from 0 to 32767. The default is 300.</p>
DACA RETRY TIME	<p>The time in seconds before a failed DACA connection is retried.</p> <p>Enter a value from 0 to 32767. The default is 600.</p>
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, DOS, Windows 3.11 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>

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

Keyword	Description
LAN AUTHORIZATION	<p>Valid entries are:</p> <p>0 LAN address authorization is not required.</p> <p>1 TME 10 Software Distribution, Version 3.1.5 for Windows NT validates LAN addresses on all LAN messages received by the TME 10 Software Distribution server.</p> <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. This LAN Authorization level has meaning in the TME 10 Software Distribution server base configuration file only.</p>
LOG FILE SIZE	<p>The maximum size of the TME 10 Software Distribution, Version 3.1.5 for Windows NT 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 uses 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 Windows NT node. The name is set up during installation and cannot be changed.</p>
MAX ATTEMPTS	<p>The maximum number of failed attempts to connect to the TME 10 Software Distribution server before the system temporarily revokes the access to the TME 10 Software Distribution server. The default is 5.</p>
MAX CONNECTIONS	<p>The maximum number of simultaneously connected local targets allowed, where 'connected' means that the target is in the process of performing a D&CC request.</p> <p>You can specify up to 100 connections. The default is 50.</p>
MAX LOCAL TARGETS	<p>The total number of local targets that can be configured for a server. The maximum number is 2000. The default is 512.</p> <p>If the single-node package is installed, this parameter must be set to 1.</p>

Editing the Base Configuration File

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

Keyword	Description
MAX REQUESTS	<p>The maximum number of requests that can be active simultaneously.</p> <p>Enter a value from 1 to 65536. The default is 8192.</p>
MAX SERVER CONNS	<p>The maximum number of connections to remote server DACAs that 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.</p> <p>Enter a value from 0 to 256. The default is 10.</p>
MAX SERVER TARGETS	<p>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.</p> <p>The maximum number is 2048. The default is 10.</p>
MAX SNADS ROUTES	<p>The maximum number of SNA/DS connections that can be defined.</p> <p>The maximum number is 800. The default is 20.</p>
MAX STS	<p>The maximum number of STS processes that can be simultaneously handling connections to remote TME 10 Software 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 <i>from</i> remote servers.</p> <p>Enter a value from 0 to 256. The default is 10.</p>
MAX STS ROUTES	<p>The maximum number of STS connections that can be defined.</p> <p>The maximum number is 1600 minus the number of SNA/DS routes that have been defined (MAX SNA/DS ROUTES). The default is 30.</p>

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 simultaneously connected user interfaces. The default is 20.</p>
MESSAGE LOG LEVEL	<p>This field defines the log level that should be used by TME 10 Software Distribution clients before they establish a connection to the TME 10 Software 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>The type of transmission protocol used. Use this keyword to specify the protocols supported on the client connection. The default is TCP/IP. If you choose to specify other protocols, enter a separate keyword for each protocol, using the following syntax.</p> <p><i><protocol name> <address></i></p> <p>Where</p> <p><i><protocol name></i></p> <p>Can be the following:</p> <ul style="list-style-type: none"> • TCP (for TCP/IP) • NBI (for NetBIOS) • IPX (for IPX/SPX) <p>The default is TCP.</p>
Continued overleaf	

Editing the Base Configuration File

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

Keyword	Description
<p>PROTOCOL (continued)</p>	<p><address></p> <p>If protocol name is TCP, enter the port number. It defaults to the TCP/IP PORT keyword in the base configuration file.</p> <p>If protocol name is NBI, specify NBI (<i>machine_name</i>) 0 10 in the NVDM.CFG file in the main installation directory.</p> <p>If protocol name is IPX, specify: IPX <i>IPX_address</i> in the NVDM.CFG file in the main installation directory.</p> <p>For a server, enter the <i>IPX_address</i> using the following syntax: <internal_network> <server_address> ► <socket></p> <p>Where: <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. <Server_address> is 000000000001 in a NetWare server. It is 12 characters long. <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 <i>IPX_address</i> using the following syntax: <network_ID> <Mac_address> <socket></p> <p>Where: <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. <Mac_address> is the physical address of the card you are using. It is 12 characters long. <socket> is 869F for a NetView DM for MVS server. The resulting string must be 24 characters long.</p>

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

Keyword	Description
REPOSITORY	<p>The name of the repository that TME 10 Software Distribution, Version 3.1.5 for Windows NT stores its objects in. The default is <prod_>\repos unless explicitly overridden. If the default repository path is changed, the repository is not removed automatically if the product is uninstalled.</p>
SERVER	<p>The name of the TME 10 Software Distribution server that serves your TME 10 Software Distribution, Version 3.1.5 for Windows NT node. 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 TME 10 Software Distribution servers, so that you are able to administer and configure them, you must name those TME 10 Software Distribution servers here. This keyword can be used up to five times. The first instance defines the TME 10 Software Distribution server that provides the TME 10 Software Distribution server function for the TME 10 Software Distribution client being defined in this table, the other instances define TME 10 Software Distribution servers that can be administered from this TME 10 Software Distribution client.</p>
SERVICE AREA	<p>The name of the directory that files and information are stored in when change files that require activation are installed. The directory should belong to a local file system mounted during startup. The default is <prod_>\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.</p>
STS IDLE TIME	<p>The time in seconds after which an idle STS connection is considered to have failed.</p> <p>Enter a value from 0 to 32767. The default is 60.</p>
STS RETRY TIME	<p>The time in seconds before a failed STS connection is retried by the Scheduler.</p> <p>Enter a value from 0 to 32767. The default is 300.</p>

Editing the Base Configuration File

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

Keyword	Description
TARGET PASSWORD AUTHENTICATION	<p>This keyword activates the target authentication feature.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • YES • NO <p>The default is NO.</p>
TCP/IP PORT	<p>The TCP/IP port number of your TME 10 Software Distribution, Version 3.1.5 for Windows NT TME 10 Software Distribution server. It must be a decimal number. This is set up automatically during the installation process and has the value 729. You need to change this field if the TCP/IP port number of your TME 10 Software Distribution server is changed. Otherwise, the field should not be altered.</p>
TRACE FILE SIZE	<p>The maximum size of the TME 10 Software Distribution, Version 3.1.5 for Windows NT internal trace files in bytes. Two trace files are used and when one is full, tracing automatically switches to the other. Internal tracing is only activated for diagnostic purposes.</p> <p>You can change this value, but it should not normally be necessary. A large value uses more disk space. A small value degrades performance very slightly, because the log is backed up more often.</p> <p>Refer to the online Command Reference for the product for information about the TME 10 Software Distribution, Version 3.1.5 for Windows NT internal tracing.</p>
WORK AREA	<p>The name of the directory that temporary work files are created and used in, and then deleted from. The default is <prod_dir>\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 Windows NT node. The name is set up during installation and cannot be changed.</p>

The following is an example of the server base configuration file:

```

# BASE CONFIGURATION FILE
#

WORKSTATION NAME:          nvdmshr
MESSAGE LOG LEVEL:        D
LAN AUTHORIZATION:        0
CONFIGURATION:            SERVER_WITH_COMMS
MACHINE TYPE:             WIN2K
LOG FILE SIZE:            500000
TRACE FILE SIZE:          1000000
API TRACE FILE SIZE:      500000
TCP/IP PORT:              729
MAX TARGETS:              600
MAX CONNECTIONS:          50
MAX USER INTERFACES:     20
SERVER:                   nvdmshr
PROTOCOL:                 TCP nvdmshr 729 50
#PROTOCOL:                NBI nvdmshr 0 50
#PROTOCOL:                IPX (NetId) (AdpAdd) (Ap1Add)
REPOSITORY:               D:\SOFTDIST\repos
SERVICE AREA:            D:\SOFTDIST\service
BACKUP AREA:              D:\SOFTDIST\backup
WORK AREA:                D:\SOFTDIST\work
INVENTORY PROGRAM:        fndinv
AUTOMATIC TARGET REGISTRATION: Y

```

Figure 32. Example of a TME 10 Software Distribution Server Base Configuration file

Changing TCP/IP Ports

TME 10 Software Distribution, Version 3.1.5 for Windows NT TME 10 Software Distribution servers listen at a TCP/IP port for TCP/IP communication from TME 10 Software Distribution clients and from other TME 10 Software Distribution servers. The following port numbers have been assigned to the product by Internet:

729	Local TME 10 Software Distribution server to TME 10 Software Distribution client communication
730, 731	Remote TME 10 Software Distribution server to TME 10 Software Distribution server communication

If in the unlikely event one of the above ports is being used by another application, and you decide to change TME 10 Software Distribution, Version 3.1.5 for Windows NT ports, the following applies:

- The local communication port, 729, is stored in the base configuration file. The port number must be changed in the base configuration file of the TME 10 Software Distribution server and the base configuration files of all its TME 10 Software Distribution clients.

Setting the LAN Authorization Level

- The remote communication ports, 729, 730, and 731, are stored in `<system_root_directory>\drivers\etc\services`. These ports have to be modified on all connected TME 10 Software Distribution servers.
 - Enter the command `echo %SYSYEMR00%` to find the system root directory.
 - Edit the file `<system root directory>\drivers\etc\services` and add the following lines:

```
netviewdm1      729/tcp
NetViewDM-snd  730/tcp
NetViewDM-rcv  731/tcp
```

Changing the TME 10 Software Distribution Server Name

When TME 10 Software Distribution, Version 3.1.5 for Windows NT is installed, a default server name and server short name are automatically defined. You can change these values to the names used in your network.

Perform the following steps:

1. Log on as *administrator*.
2. Stop all activity on the server and on clients by entering the following command on the workstations:

```
nvdm stop -k -x
```
3. Rename the server target using the `nvdm rentg` command at the command line interface (see the online Command Reference for the product).
4. Edit the `WORKSTATION NAME` and `SERVER` keywords in the `nvdm.cfg` file to reflect the server's new name. See "Base Configuration File Parameters" on page 75.
5. Edit the name of the server in `nvdm.cfg` on each client to reflect the new name of the server.
6. On both server and client delete the contents of the `uicfg` directory
7. Start the server and the clients by entering the following command at the workstations:

```
nvdm start
```

Setting the LAN Authorization Level

You can change the level of checking that is applied to LAN messages received from TME 10 Software Distribution clients.

Messages received by the TME 10 Software Distribution server identify their origin using TCP/IP name and LAN address information. The messages also carry the burned-in LAN address for the LAN adapter on the TME 10 Software Distribution client. You can ensure that the TME 10 Software Distribution server checks this burned-in address with a value configured at the TME 10 Software Distribution server

Setting the LAN Authorization Level

for the TME 10 Software Distribution client. This is an extra security measure that helps to prevent unauthorized use of the LAN.

Use the graphical interface to set the level of LAN adapter checking applied by TME 10 Software Distribution, Version 3.1.5 for Windows NT as follows:

1. Access the main Catalog window of the graphical interface.
2. Choose **System** from the menu bar.
3. Select **Validate LAN** from the menu.

A cascade menu is displayed showing **On** and **Off**. The current setting cannot be selected. To change the setting, select the entry that is available. Cancel the menu to retain the current setting.

You can also set the LAN authorization level for the current session using the **updb**s command from the command line interface (see the online Command Reference for the product).

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 Windows NT 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 Windows NT, 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.

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.

Defining Targets

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
<p>Target address</p>	<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 Windows NT 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:</p> <p><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>

Defining Targets

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 95.</p> <p>Server</p> <p>Targets running TME 10 Software Distribution, Version 3.1.5 for Windows NT 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
<p>Target type (cont.)</p>	<p>Single</p> <p>Targets running TME 10 Software Distribution, Version 3.1.5 for Windows NT 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 Windows NT 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> <ul style="list-style-type: none"> 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)
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 Windows NT 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 Windows NT target If the target you are defining is <i>not</i> a TME 10 Software Distribution, Version 3.1.5 for Windows NT 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	<p>The password used to access the target. It can have from 6 to 8 characters.</p>

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 76), 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 Windows NT servers to NetView DM/2 servers or NetView DM for MVS focal points.

Chapter 12. Configuring STS Remote Connection Files

This chapter describes how to configure the connection files required to define an STS (Server-to-Server) network using 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 held in three different text files, which can be edited using a text editor.

The files that can be edited are:

STS configuration file

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

STS connection configuration files

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

Routing table

This file instructs the TME 10 Software Distribution server which 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 Windows NT is running, you must use the **rld** command to reload configuration changes. See the online *Command Reference* manual for more information about this command.

STS Configuration File

The STS configuration file, `<prod_dir>\db\snadscfg`, contains a parameter for controlling the use that TME 10 Software Distribution, Version 3.1.5 for Windows NT makes of the STS network, **ORIGIN HOP COUNT**.

There can be only one configuration file at a TME 10 Software 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 105). If you have a mixed network, specify all the keywords in a single file.

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

STS Connection Configuration File

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 hash character (#).

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 Windows NT when the hop count is omitted or is specified as zero in the routing table. (See "Editing the STS Routing Table" on page 99.) An appropriate value for this field is 5.

The following is an example of the STS Configuration File.

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

Figure 33. Example of an STS configuration file

STS Connection Configuration File

STS connection configuration files define the details of 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 <prod_dir>\db\snadscon, where <prod_dir> is the directory where the product was installed. A sample file named CONNSTS is provided.

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 15 on page 99. 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 hash character (#).

<i>Table 15. STS Connection Configuration File Parameters</i>	
Keyword	Description
PROTOCOL	Specify: TCP/IP The TCP/IP protocol is used. IPX The IPX protocol is used. NBI The NetBIOS protocol is used.
TYPE	Specify STS for a server-to-server connection type.
REMOTE SERVER NAME	The workstation name of the remote server.

The following is an example of an STS connection configuration file for the TCP/IP connection protocol. This sample file, called CONNTCP, is installed during product installation. Note that the value for the PROTOCOL keyword can be either APPC or TCP/IP, depending on which communication protocol you are using.

```
# SNA/DS CONNECTION CONFIGURATION FILE FOR CONNECTION connststcp
#
# This connection is used to handle transmissions between
# TME 10 Software Distribution for Windows NT servers using STS across TCP/IP.
#
# This file should be stored as <prod_dir>\db\snadscfg

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

Figure 34. Example of an STS connection configuration file for TCP/IP

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 Windows NT server which 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 installed. 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.

Editing the STS Routing Table

A sample routing table is set up for you when you install the communication option. 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 TME 10 Software Distribution server and for each of the TME 10 Software Distribution clients in a network. It also defines a number of routes. Each route has the following data associated with it:

- Protocol type: APPC or TCP/IP or BOTH
- 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 server is attached to just one remote node. Only one connection is defined and this takes all distributions to and from the TME 10 Software 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 (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 Windows NT network. To identify a target's <domain address> and <target address> look for these corresponding terms:

Table 16. Target Identification Terms

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.

To define a target using the command line interface, see the *Software Distribution for Microsoft Windows NT® Reference*.

Defining Routes

Before defining the routes in a routing table, you must specify the type of network you are defining. 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 (#).

Editing the STS Routing Table

Note: In an STS network, if you do *not* need an SNA/DS connection, do not select either APPC or BOTH (APAR I11831), to avoid running the Transmission Control Program unnecessarily. Instead, use whichever of the other protocols is used by the clients in in the LAN.

Enter the following information in the order shown to define a route.

Parameter	Description
Destination Address	<p>The destination nodes that this route serves.</p> <p>TME 10 Software Distribution, Version 3.1.5 for Windows NT 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 100 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 <prod_dir>\db\sna dscon directory (see "Editing STS Connection Configuration Files" on page 97). 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 97).</p>

A sample routing table follows. It is supplied with TME 10 Software Distribution, Version 3.1.5 for Windows NT in the <prod_dir>\db\rou tetab file. The comment line

containing the column headings for the configuration information is included to facilitate reading.

An example of an STS routing table for TCP/IP routes follows.

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

NETWORK PROTOCOL:  TCP/IP

#
# Destination      Connection      Hop
# Address          Count
#
SRVTCIP1.*        CONNTCP1        10
SRVTCIP2.*        CONNTCP2        10
*.*               CONNTCP3        10
```

Figure 35. Example of an STS routing table for TCP/IP routes

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:

- **Server name and target address**

You provide this information when you define a target using the graphical interface or the **addtg** line command (see the online *TME 10 Software Distribution for Windows NT Command Reference*). The combination of the two (<server name>.<target address>) uniquely identifies the target in a network.

Chapter 13. 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 Windows NT 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 Windows NT'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 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 Windows NT is running, you must use the **rld** command to reload configuration changes. See the *TME 10 Software Distribution for Windows NT Reference* for more information about this command.

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 Windows NT 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 17 on page 106. 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 hash character (#).

SNA/DS Configuration File

<i>Table 17. SNA/DS Configuration File Parameters</i>	
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 Windows NT when the hop count is omitted or is specified as zero in the routing table. (See "Editing the SNA/DS Routing Table" on page 110.) 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 Windows NT sends in one go 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 Windows NT after a severe transitory error or an error requiring operator intervention has been detected.</p> <p>The value you enter should be 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 Windows NT 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>The value entered should be 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\snaadscfg

ORIGIN HOP COUNT:           10
MAX CRMU:                   5
TRANSMISSION HOLD TIME:    3600
ALLOCATION FAILURE RETRY TIME: 1000
```

Figure 36. 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\connsna, where <prod_dir> is the directory where the product was installed. A sample file, named CONNSNA, is provided for APPC connections over SNA/DS.

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 18. 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 hash character (#).

Table 18 (Page 1 of 3). SNA/DS Connection Configuration File Parameters

Keyword	Description
PROTOCOL	Specify: APPC Only the APPC protocol is available.
TYPE	Specify: SNA Only an SNA/DS connection is available.

SNA/DS Connection Configuration File

Table 18 (Page 2 of 3). SNA/DS Connection Configuration File Parameters

Keyword	Description
<p>SEND TP SYMBOLIC DESTINATION</p>	<p>The name of the MS SNA Server Side Information Profile that is used when TME 10 Software Distribution, Version 3.1.5 for Windows NT starts a conversation to receive on this connection. That is, the profile to use when starting a conversation with the Send TP at the remote node. The name of the profile is assigned when you configure SNA profiles. The entry can be up to eight characters long.</p> <p>This keyword is valid only if the PROTOCOL specified is APPC.</p>
<p>RECEIVE TP SYMBOLIC DESTINATION</p>	<p>The name of the MS SNA Server Side Information Profile that is to be used when TME 10 Software Distribution, Version 3.1.5 for Windows NT starts a conversation to Send on this connection. That is, the profile to use when starting a conversation with the Receive TP at the remote node. The name of the profile is assigned when you configure SNA profiles. The entry can be up to eight characters long.</p> <p>This keyword is valid only if the PROTOCOL specified is APPC.</p>
<p>REMOTE SERVER NAME</p>	<p>The workstation name of the remote server. TCP/IP must be able to resolve this host name to an IP address, therefore the name must be defined in the /etc/hosts directory or on the domain name server.</p> <p>This keyword is valid only if the PROTOCOL specified is TCP/IP.</p>
<p>TCP/IP TIME-OUT</p>	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for Windows NT waits for a message on a TCP/IP connection before deciding there is a communication error and thus terminates the distribution.</p> <p>This keyword is valid only if the PROTOCOL specified is TCP/IP. The value entered should be a decimal number. A reasonable value is 300 (five minutes).</p>
<p>NEXT DSU</p>	<p>The RGN and REN 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. If the PROTOCOL specified is TCP/IP, this field must specify the target address and the domain address of the remote server.</p> <p>The RGN and REN must be entered as consecutive strings separated by a period.</p>

Table 18 (Page 3 of 3). SNA/DS Connection Configuration File Parameters

Keyword	Description
TRANSMISSION TIME-OUT	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for Windows NT 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>The value entered should be a decimal number. A reasonable value is 60 seconds.</p>
RETRY LIMIT	<p>The number of times that TME 10 Software Distribution, Version 3.1.5 for Windows NT is to retry the transmission of a distribution (file or change management command) before reporting an error.</p> <p>The value entered should be a decimal number. A reasonable value is 3.</p>
SEND MU_ID TIME-OUT	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for Windows NT 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>The value entered should be a decimal number. A reasonable value is 60.</p>
RECEIVE MU_ID TIME-OUT	<p>The number of seconds that TME 10 Software Distribution, Version 3.1.5 for Windows NT is to wait before retrying the receipt of a distribution. This value prevents thrashing if a transitory error has been detected.</p> <p>The value entered should be a decimal number. A reasonable value is 120.</p>

The following is an example of an SNA/DS connection configuration file for the APPC protocol. This sample file, called CONNSNA, is installed during product installation.

Editing the SNA/DS Routing Table

```
# SNA/DS CONNECTION CONFIGURATION FILE FOR CONNECTION CONNSNA
#
# This connection is used to handle transmissions between
# TME 10 Software Distribution, Version 3.1.5 for Windows NT servers.
#
# This file should be stored as \<prod_dir>\db\snacon\connsna

PROTOCOL:                APPC
TYPE:                    SNA
SEND TP SYMBOLIC DESTINATION:  NVDMSIDS
RECEIVE TP SYMBOLIC DESTINATION: NVDMSIDR
NEXT DSU:                NTRK1.LU0001
TRANSMISSION TIME-OUT:    60
RETRY LIMIT:             3
SEND MU_ID TIME-OUT:     60
RECEIVE MU_ID TIME-OUT:  120
```

Figure 37. 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 have remote communication installed. 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 install the communication option. 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.

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.

Editing the SNA/DS Routing Table

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 Windows NT network. To identify a target's <domain address> and <target address> look for these corresponding terms:

Table 19. Target Identification Terms

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 hash character (#).

Enter the following information in the order shown to define a route.

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 Windows NT 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: <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 111 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 Windows NT 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 Windows NT 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>

Editing the SNA/DS Routing Table

Parameter	Description
<p>Distribution Protection</p>	<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>
<p>Maximum Size Distribution</p>	<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 1MB of data are sent on this connection.</p> <p>4MEGABYTES Only distributions with less than 4MB 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>

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 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 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 105).</p>

Sample routing tables follow. They are supplied with TME 10 Software Distribution, Version 3.1.5 for Windows NT in the \<prod_dir>\db\routetab file, where <prod_dir> is the directory where the product was installed. The comment line containing the column headings for the configuration information is included to facilitate reading.

Editing the SNA/DS Routing Table

```
# SNA/DS ROUTING TABLE
# This table provides SNA/DS routing information for a
# TME 10 Software Distribution, Version 3.1.5 for Windows NT SNA network.
# This file should be stored as \<prod_dir>\db\routetab

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 38. Example of an SNA/DS routing table for APPC routes

An example of an SNA/DS routing table for the APPC connection protocol follows:

```
# SNA/DS ROUTING TABLE
# This table provides SNA/DS routing information for
# a TME 10 Software Distribution combination TCP/IP and APPC network.
# This file should be stored as \<prod_dir>\db\routetab

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
SERVER02.SERVER02  ANY  ANY  ANY  ANY  SERVER01
SERVER02
```

Figure 39. Example of an SNA/DS routing table for both APPC and TCP/IP protocols

Chapter 14. Configuring Microsoft SNA Server

This chapter explains how to configure the Microsoft SNA Server. You must configure the Microsoft SNA Server to accomplish the following tasks:

- To enable TME 10 Software Distribution, Version 3.1.5 for Windows NT or Windows 2000 to communicate with NetView DM for MVS
- To enable you to establish an APPC connection between the TME 10 Software Distribution for Windows NT or Windows 2000 server and another TME 10 Software Distribution, Version 3.1.5 server

The Microsoft SNA Server documentation contains additional information about how to configure the SNA Server product.

Prerequisites

The prerequisites depend on whether your server has Windows NT or Windows 2000 installed:

Windows NT SNA Server 4.0 or later

Windows 2000 SNA Server 4.0 with Service Pack 3, or later

General Assumptions

For this scenario it is assumed that:

- The workstation is connected with the host via Token Ring
- Local LU is I9RLM000
- CP Name is I9RLM000
- Network name is ITIBM0PC
- Local Node ID is 05D-80000
- Partner LU is N161TC04
- Partner CP name is N161TC04
- Partner Network name is ITIBM0PC

Configuring SNA Server Parameters

To configure SNA Server parameters, perform the following steps:

- 1** Select **Start ▶ Programs ▶ Microsoft SNA Server ▶ Manager**. The Microsoft SNA Server Manager window appears.

Configuring Microsoft SNA Server

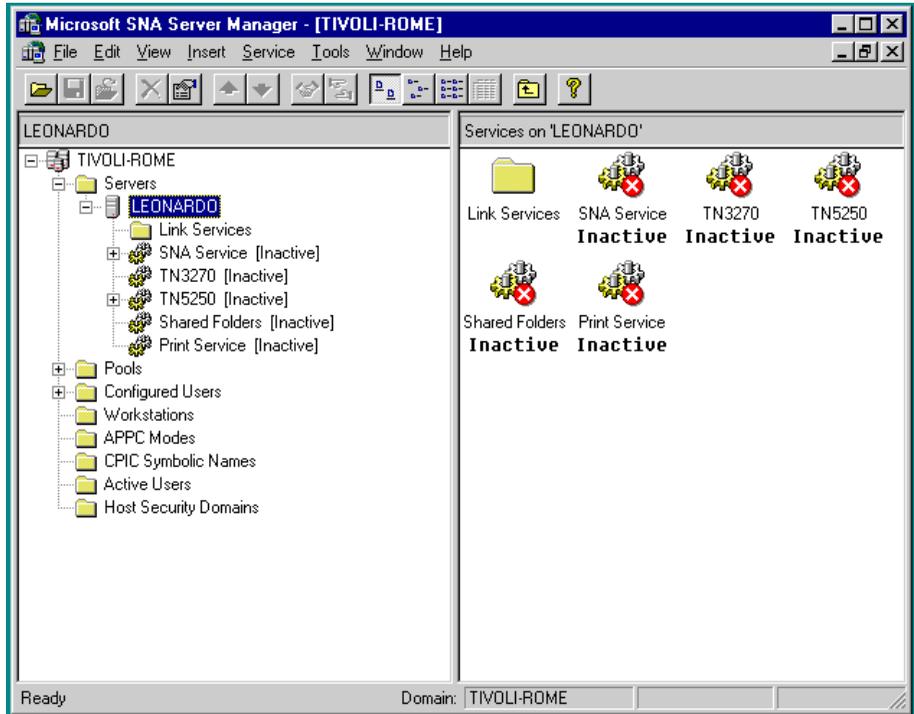


Figure 40. Microsoft SNA Server Manager

- 2 From the Microsoft SNA Server Manager window select the server, as shown in Figure 40.
- 3 Click the plus sign next to the server to expand the list and select **SNA Service**, as shown in Figure 41 on page 119.

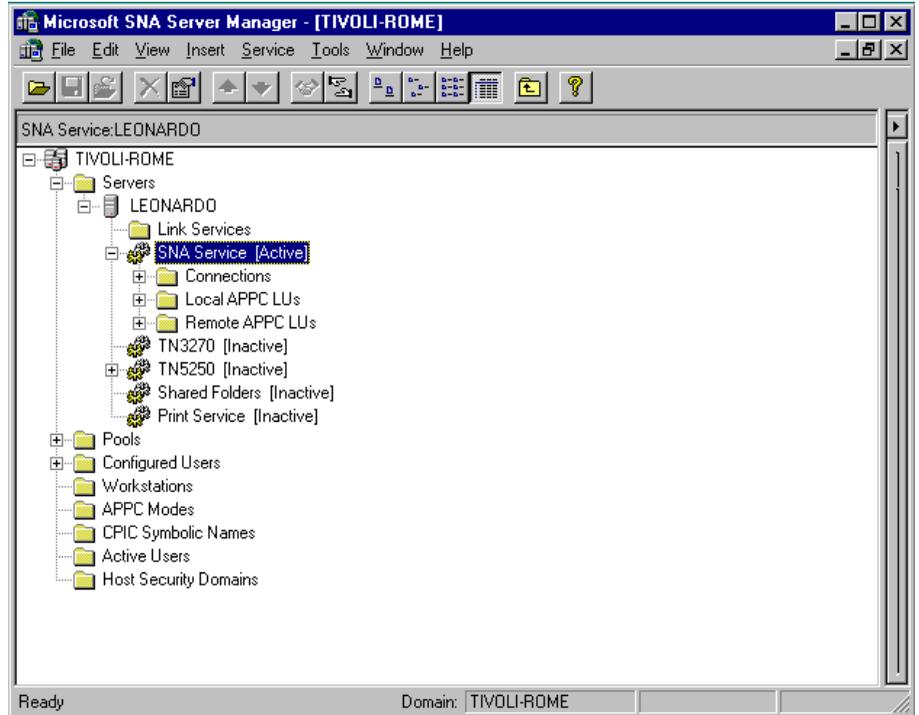


Figure 41. Selecting SNA service

The SNA Service: Server Properties window appears.

Configuring LU Properties

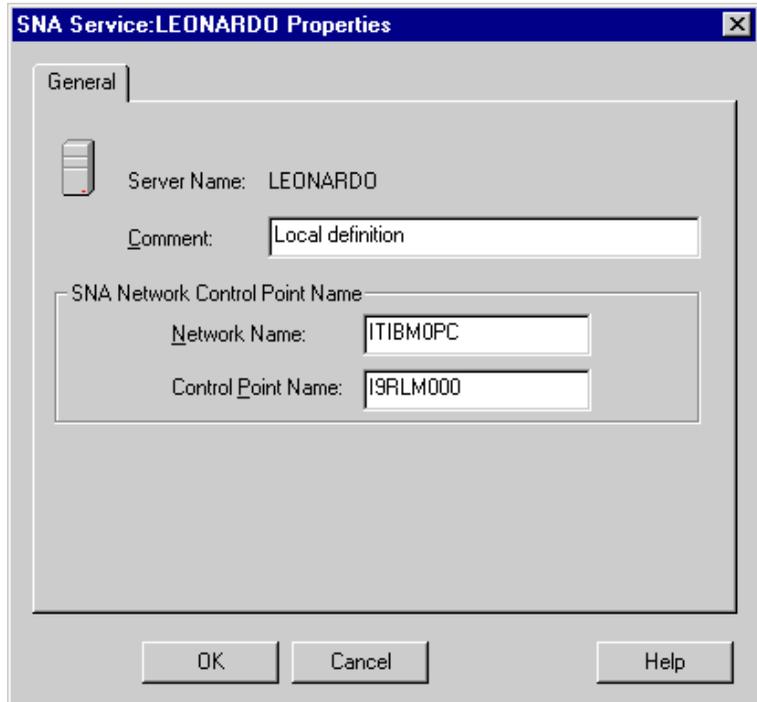


Figure 42. Configuring network and control point names

- 4 On the General page, in the SNA Network Control Point Name group box, enter the network name and control point name of your TME 10 Software Distribution for Windows NT server, as shown in Figure 42.

The network name identifies the SNA network of the local SNA server. The control point name identifies the local system to the other control points on the SNA network. These names can be from one to eight characters long.

- 5 Optionally, type a comment. The comment can be from one to 25 characters long.
- 6 Click **OK**.

Configuring LU Properties

To configure LU properties, perform the following steps:

- 1 From the Microsoft SNA Server Manager window, select **Local APPC LUs**, as shown in Figure 43 on page 121.

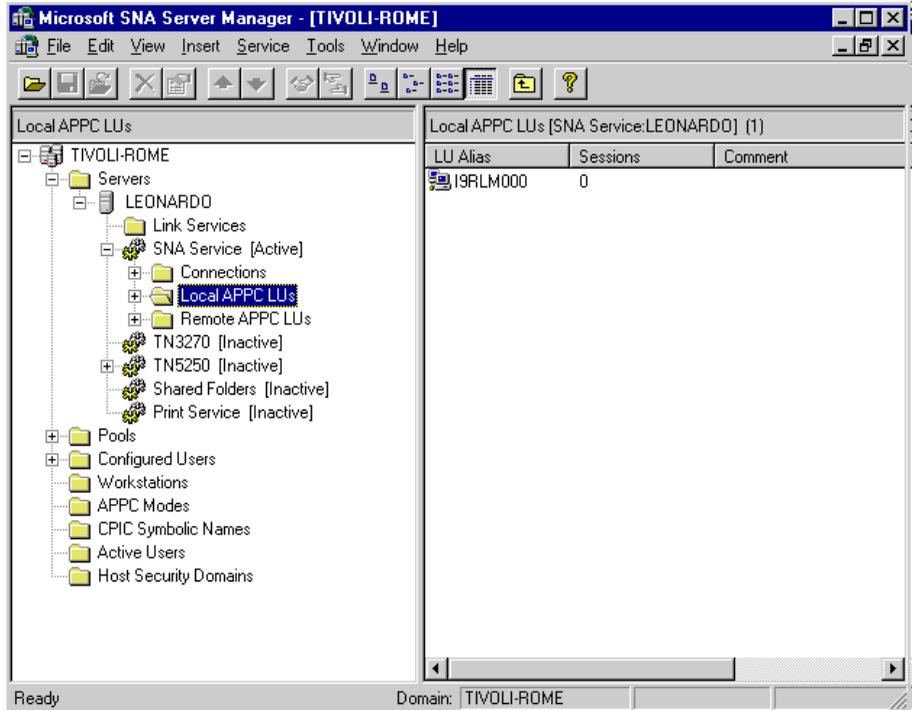


Figure 43. Selecting local APPC LUs

- 2 Right-click **Local APPC LUs** and select **Insert ► APPC ► Local LU**, as shown in Figure 44 on page 122.

Configuring LU Properties

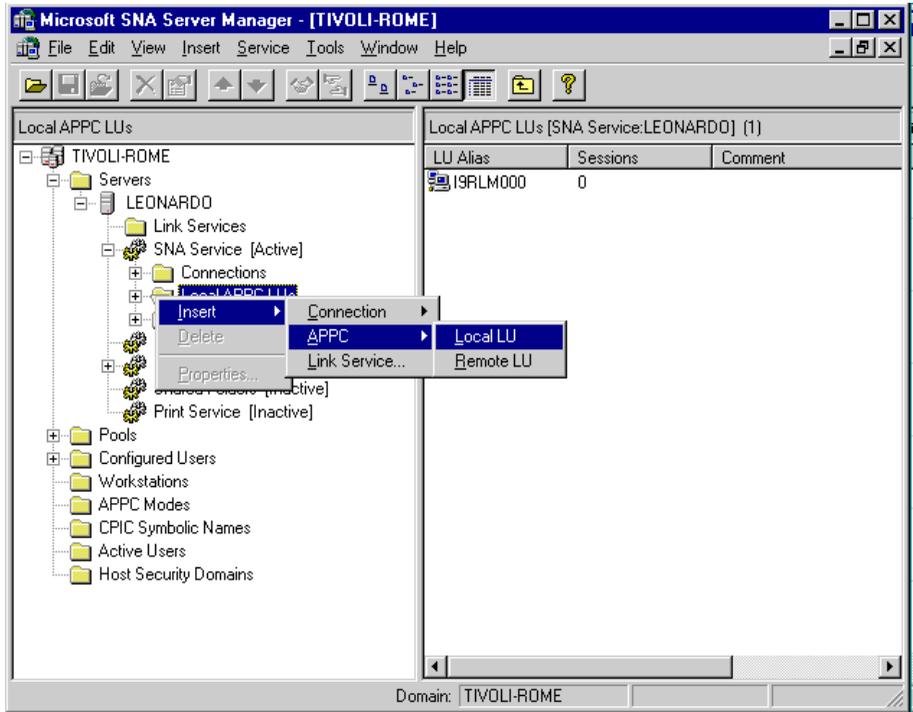


Figure 44. Selecting an LU name

The APPC LU Properties window appears.

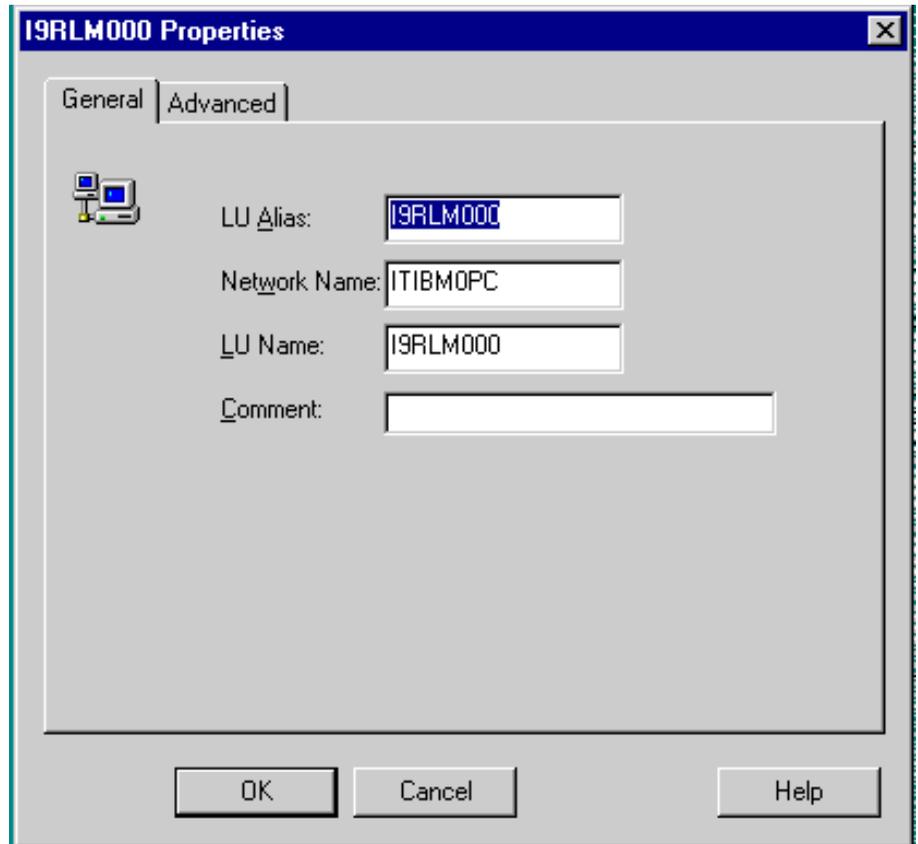


Figure 45. Configuring APPC LU properties: general page

- 3 On the General page, enter the following information:
 - a. In the **LU Alias** field, enter the LU alias which identifies the LU to local TPs. It can be the same as the LU name, as shown in Figure 45. The alias can be from one to eight characters long.
 - b. In the **Network Name** field, type the network name It can be from one to eight characters long.
 - c. In the **LU Name** field, type the LU Name It can be from one to eight characters long.
 - d. Optionally, type a comment. It can be from one to 25 characters long.
- 4 Select the **Advanced** tab. The Advanced page appears.

Configuring APPC Modes

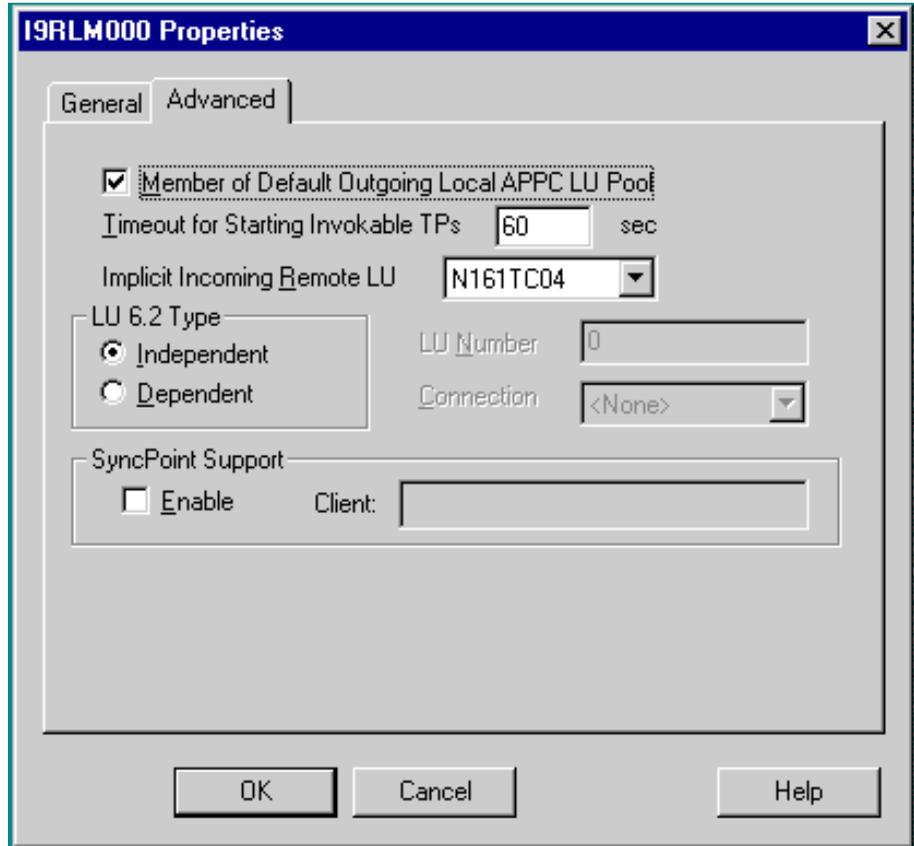


Figure 46. Configuring APPC LU properties: advanced page

- 5 On the Advanced page, check **Member of Default Outgoing Local APPC LU Pool**.
- 6 Enter **60**. as the **Timeout for Starting Invokable TPs** field.
- 7 From the Implicit Incoming Remote LU drop-down list, select **N161TC04**.
- 8 In the LU 6.2 Type box, select the **Independent** option button.
- 9 Click **OK**.

Configuring APPC Modes

To configure APPC modes, perform the following steps:

- 1 From the Microsoft SNA Server Manager window, select **APPC Modes ► Insert ► APPC ► Mode Definition**, as shown in Figure 47 on page 125.

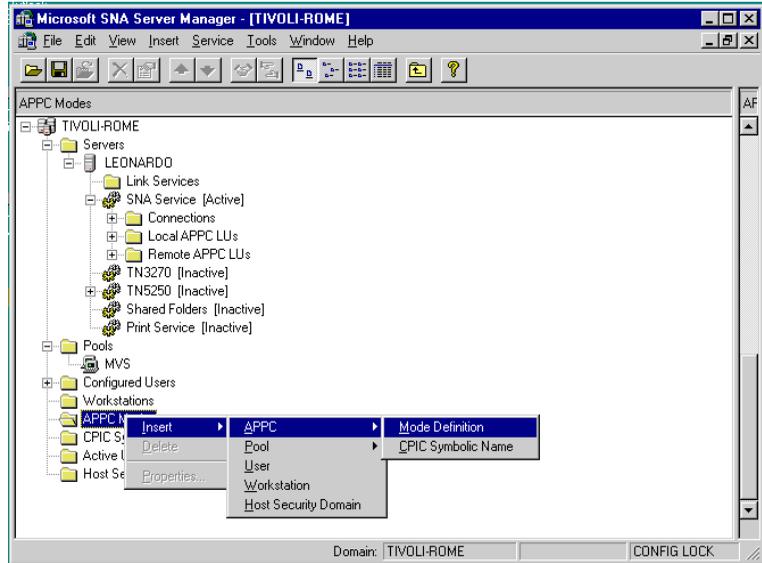


Figure 47. Selecting APPC modes

The APPC Mode Properties window appears.

Configuring APPC Modes

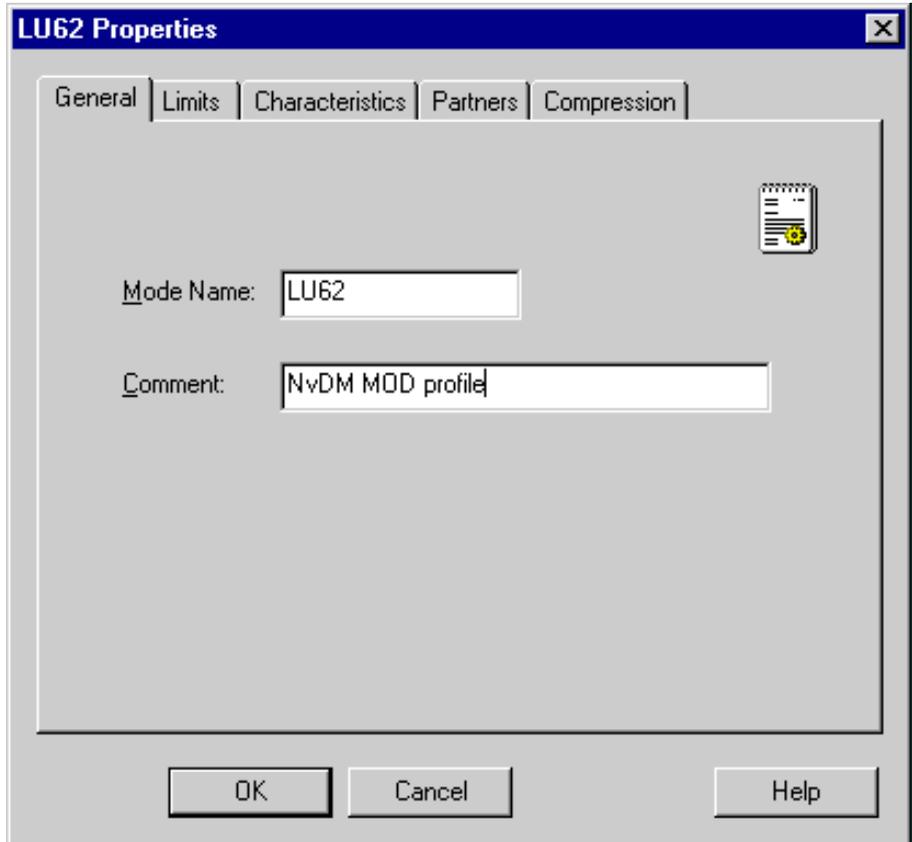


Figure 48. Configuring APPC modes: general page

- 2 On the General page, enter the **Mode Name**.
- 3 Optionally, type a comment. It can be from one to 25 characters long.
- 4 Select the **Limits** tab. The Limits page appears.

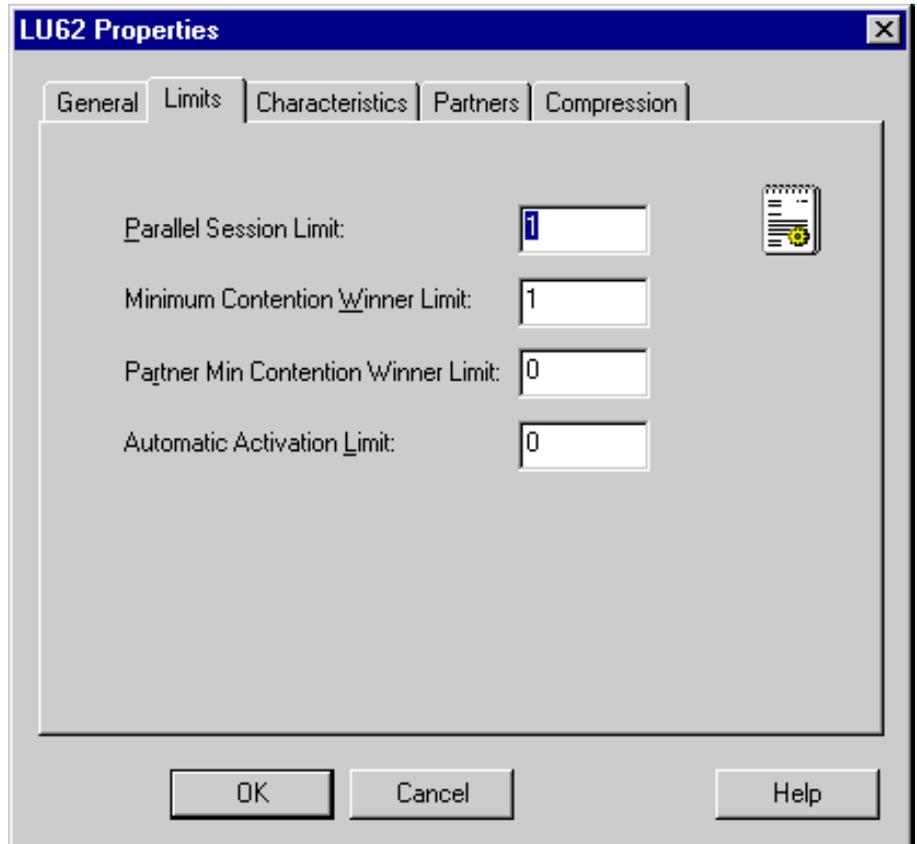


Figure 49. Configuring APPC modes: limits page

- 5 Keep the default value for the following fields, as shown in Figure 49:
 - **Parallel Session Limit**
 - **Minimum Contention Winner Limit**
 - **Partner Min Contention Winner Limit**
 - **Automatic Activation Limit**
- 6 Select the **Characteristics** tab. The Characteristics page appears.

Configuring a Downstream Connection

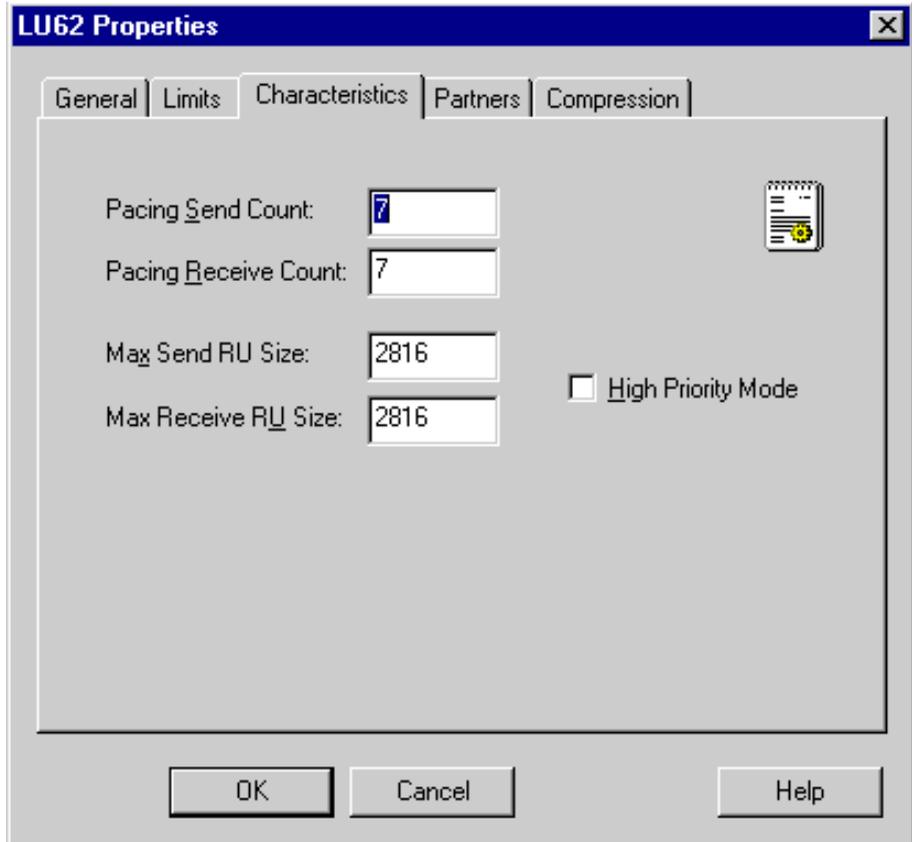


Figure 50. Configuring APPC modes: characteristics page

- 7 Keep the default value for the following fields, as shown in Figure 50:
 - **Pacing Send Count**
 - **Pacing Receive Count**
 - **Max Send RU Size**
 - **Max Receive RU Size**
- 8 In the Partners and Compression pages, leave the default values.
- 9 Click **OK**.

Configuring a Downstream Connection

To configure a downstream connection, perform the following steps:

- 1 From the Microsoft SNA Server Manager window, select **Connections ► Insert ► Connection ► 802.2**, as shown in Figure 51 on page 129.

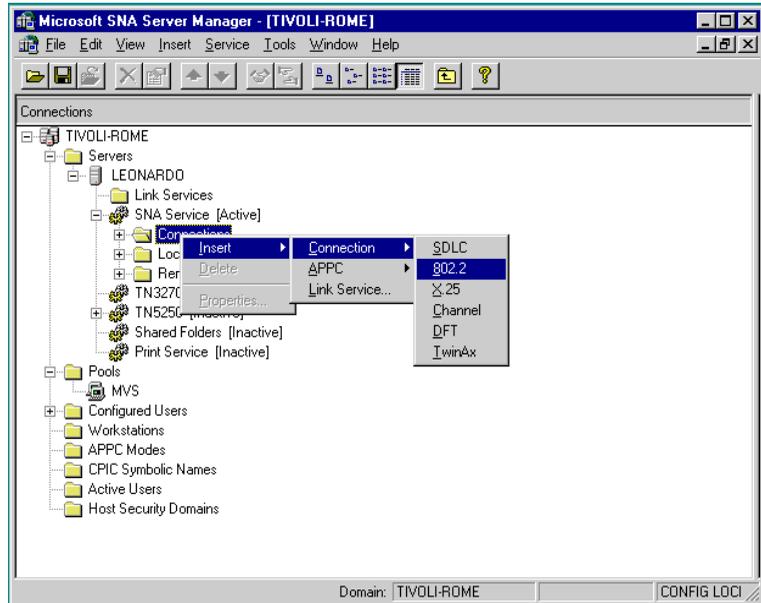


Figure 51. Selecting a downstream connection

The Connection Properties window appears.

Configuring a Downstream Connection

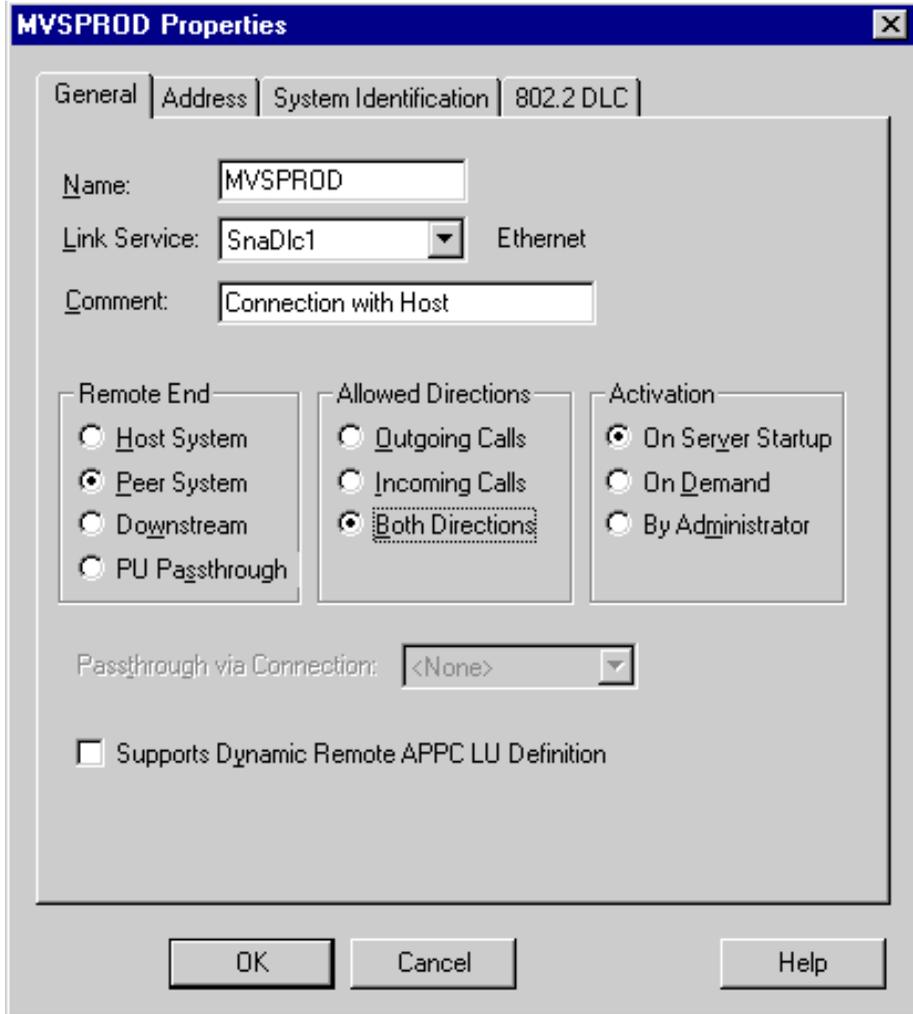


Figure 52. Configuring connection properties: general page

- 2 On the General page, enter a connection name in the **Connection name** field, as shown in Figure 52.
- 3 From the Link Service drop-down list, select **SnaDlc1**. (the DLC protocol must already be installed)
- 4 Optionally, fill in the **Comment** field.
- 5 Select the option button in the following group boxes, as shown in Figure 52:
 - **Remote End**
 - **Allowed Directions**
 - **Activation**

6 Select the **Address** tab. The Address page appears.

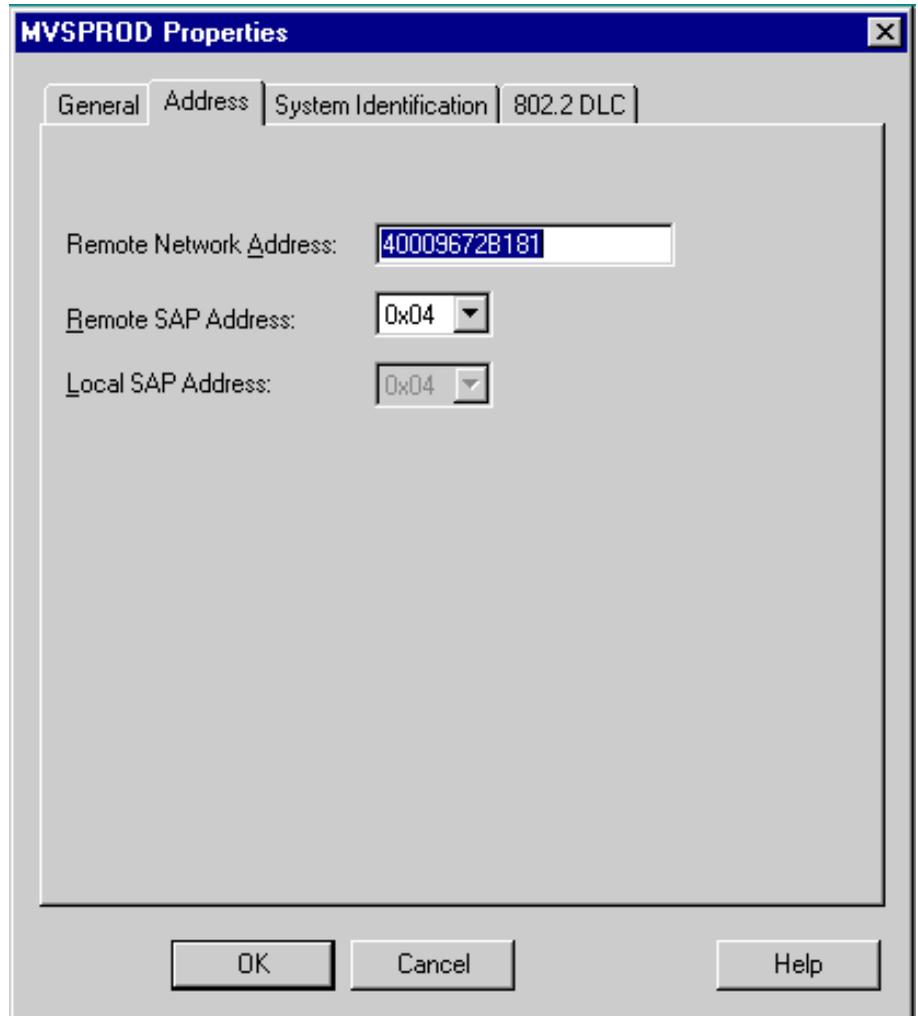


Figure 53. Configuring connection properties: address page

7 Enter the following fields, as shown in Figure 53:

- **Remote Network Address**

The 12-digit hexadecimal network address of the remote host, peer, or downstream system.

- **Remote SAP Address**

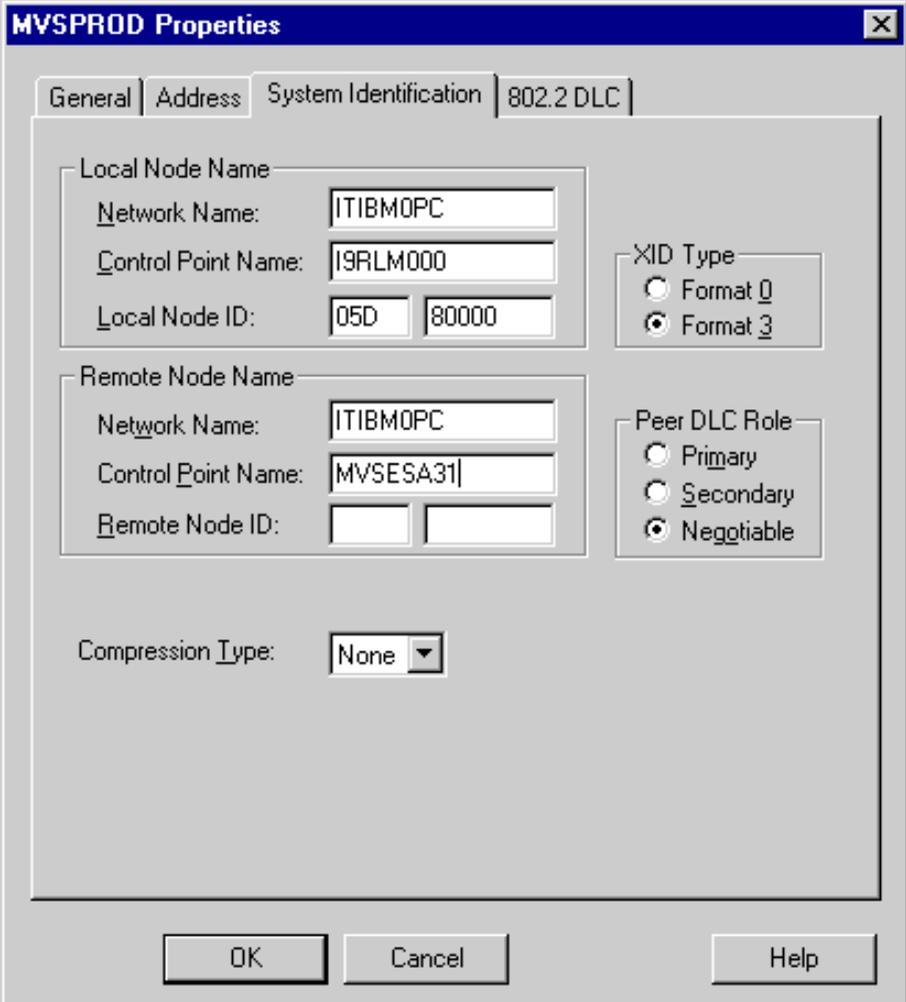
A two-digit hexadecimal number that is a multiple of four, between 04 and EC. See your token ring or Ethernet manual for more information.

- **Local SAP Address**

Configuring a Downstream Connection

A two-digit hexadecimal number that is a multiple of four, between 04 and EC. See your token ring or Ethernet manual for more information.

- 8 Select the **System Identification** tab. The System Identification page appears.



The screenshot shows the 'MVSPROD Properties' dialog box with the 'System Identification' tab selected. The '802.2 DLC' sub-tab is also active. The 'Local Node Name' group contains three text boxes: 'Network Name' (ITIBM0PC), 'Control Point Name' (I9RLM000), and 'Local Node ID' (05D and 80000). The 'Remote Node Name' group contains three text boxes: 'Network Name' (ITIBM0PC), 'Control Point Name' (MVSESA31), and 'Remote Node ID' (empty). The 'Compression Type' is set to 'None'. The 'XID Type' section has 'Format 3' selected. The 'Peer DLC Role' section has 'Negotiable' selected. Buttons for 'OK', 'Cancel', and 'Help' are at the bottom.

Figure 54. Configuring connection properties: system identification page

- 9 In the Local Node Name group box, enter your values as shown in Figure 54 for the following text boxes:
 - a. **Network Name**
It can be from one to eight characters long.
 - b. **Control Point Name**
It can be from one to eight characters long.

Configuring Remote LU Properties

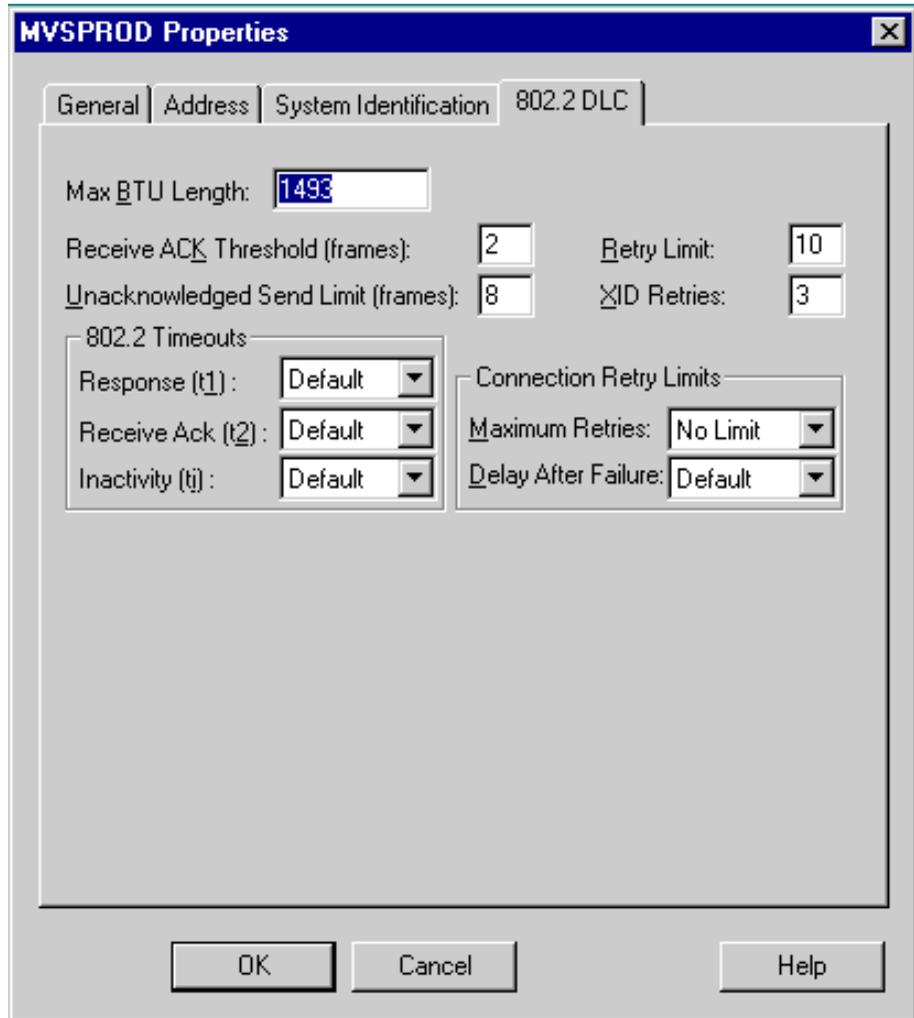


Figure 55. Configuring APPC modes: 802.2 DLC page

- 14 On the 802.2 DLC page, use the same values that are shown in Figure 55.
- 15 Click **OK**.

Configuring Remote LU Properties

To configure remote LU properties, perform the following steps:

- 1 From the Microsoft SNA Server Manager window, select **Remote APPC LUs** ► **Insert** ► **APPC** ► **Remote LU**, as shown in Figure 56 on page 135.

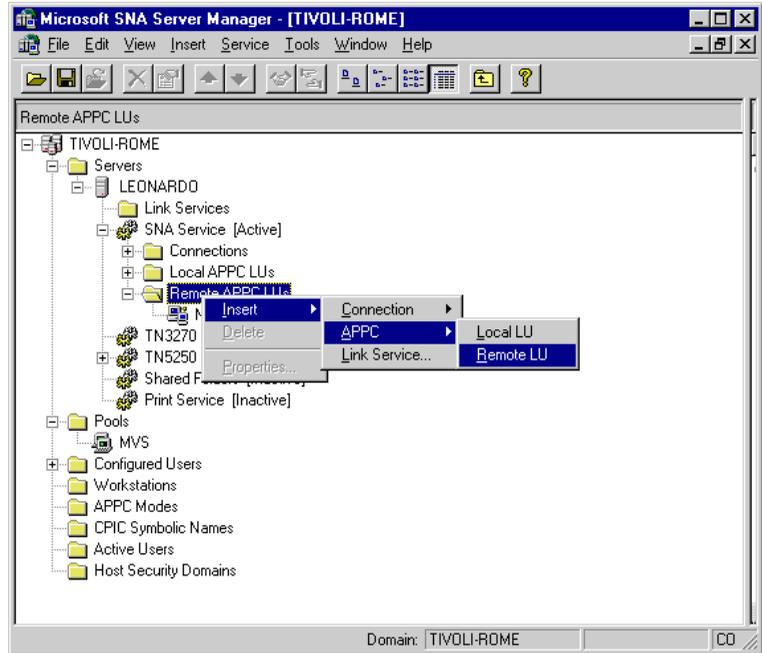


Figure 56. Configuring remote LU properties

The Connection Properties window appears.

Configuring Remote LU Properties

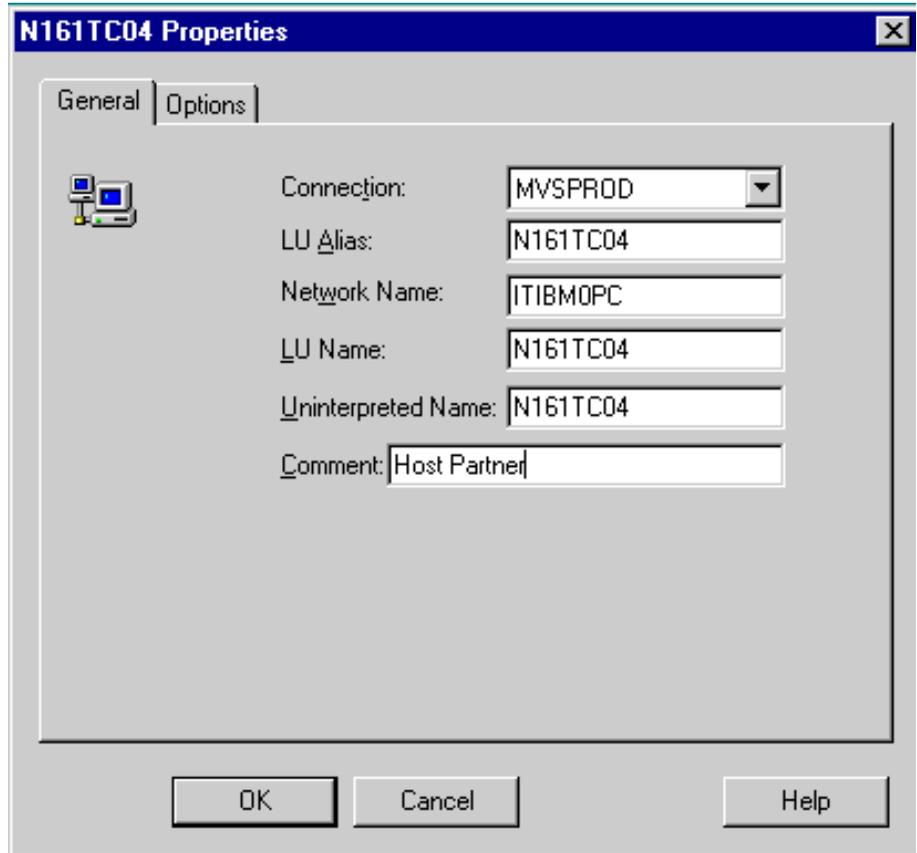


Figure 57. Configuring remote LU properties: general page

2 On the General page, enter your values in the following fields, as shown in Figure 57:

- **Connection**

It identifies the connection type.

- **LU Alias**

It identifies the LU to local TPs. The LU Alias can be from one to eight characters long, and it can be the same as the LU Name.

- **Network Name**

It identifies the SNA network to the remote LU. The Network Name can be from one to eight characters long.

- **LU Name**

It can be from one to eight characters long.

- **Uninterpreted Name**

It can be from one to eight characters long.

- **Comment**

It is optional. Comments can be from one to 25 characters long.

3 Select the **Options** tab.

The Options page appears.

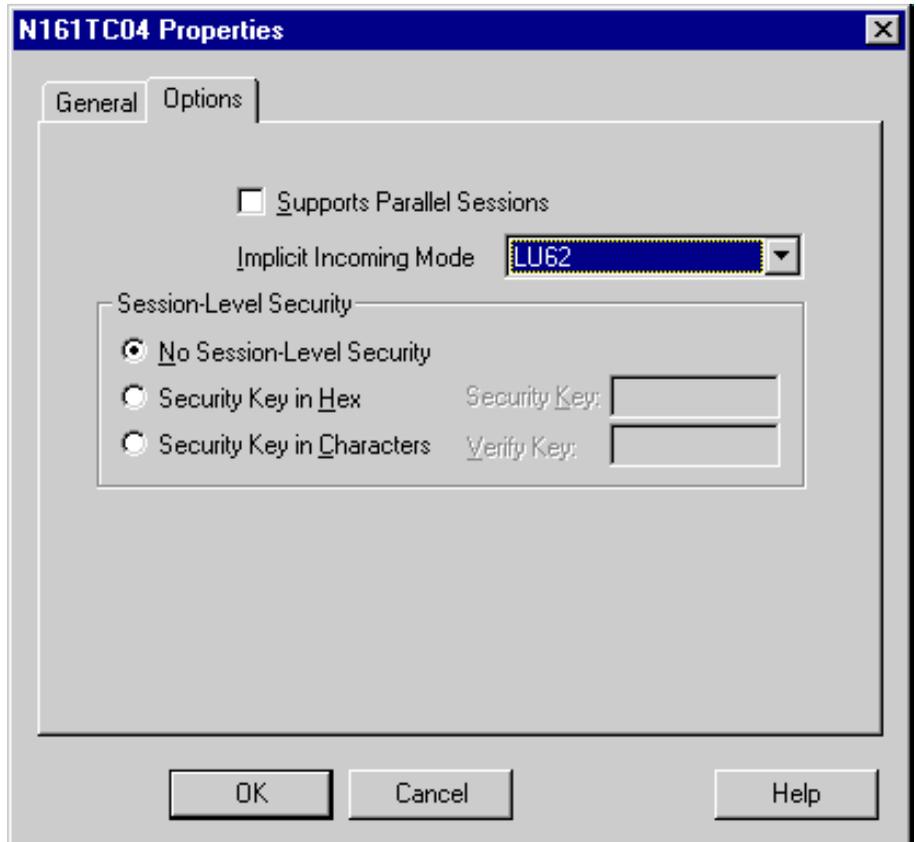


Figure 58. Configuring remote LU properties: options page

4 On the Options page, deselect **Supports Parallel Sessions**.

5 Select **LU62** from the **Implicit Incoming Mode** text box as shown in Figure 58.

6 In the Session-Level Security group box, check the **No Session-Level Security** option button.

7 Select **OK**.

Creating CPI-C Information Records

To create a CPI-C information records, perform the following steps:

- 1 From the Microsoft SNA Server Manager window, select **CPIC Symbolic Names** ► **Insert** ► **APPC** ► **CPIC Symbolic Name**, as shown in Figure 59.

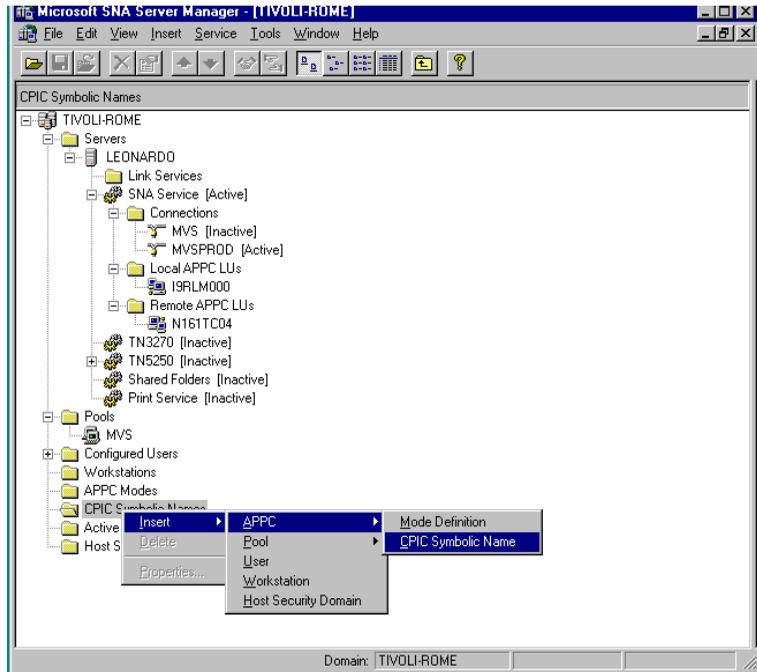


Figure 59. Selecting a CPIC symbolic name

The CPIC Properties window appears.

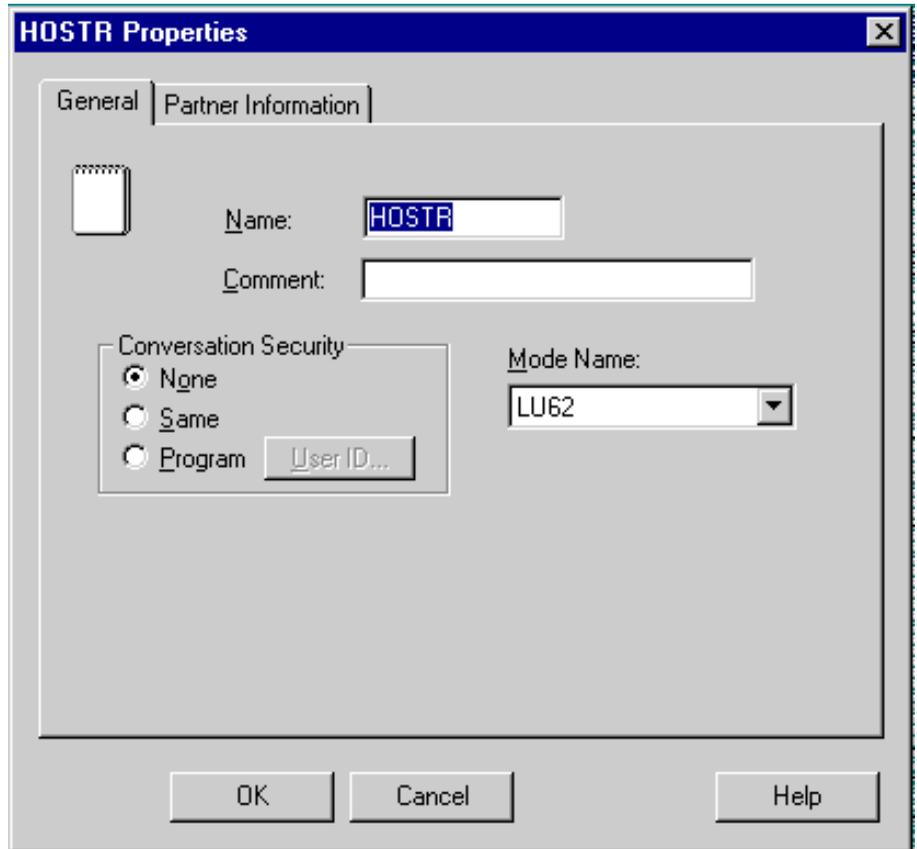


Figure 60. Configuring the CPIC properties: general page for HOSTR

- 2 On the General page, enter **HOSTR** in the Name field, as the CPI-C Symbolic Destination Name, as shown in Figure 60.
- 3 Click the **None** option button in the Conversation Security group box
- 4 Select **LU62** in the Mode Name drop-down list.
- 5 Select the **Partner Information** tab. The Partner Information page appears.

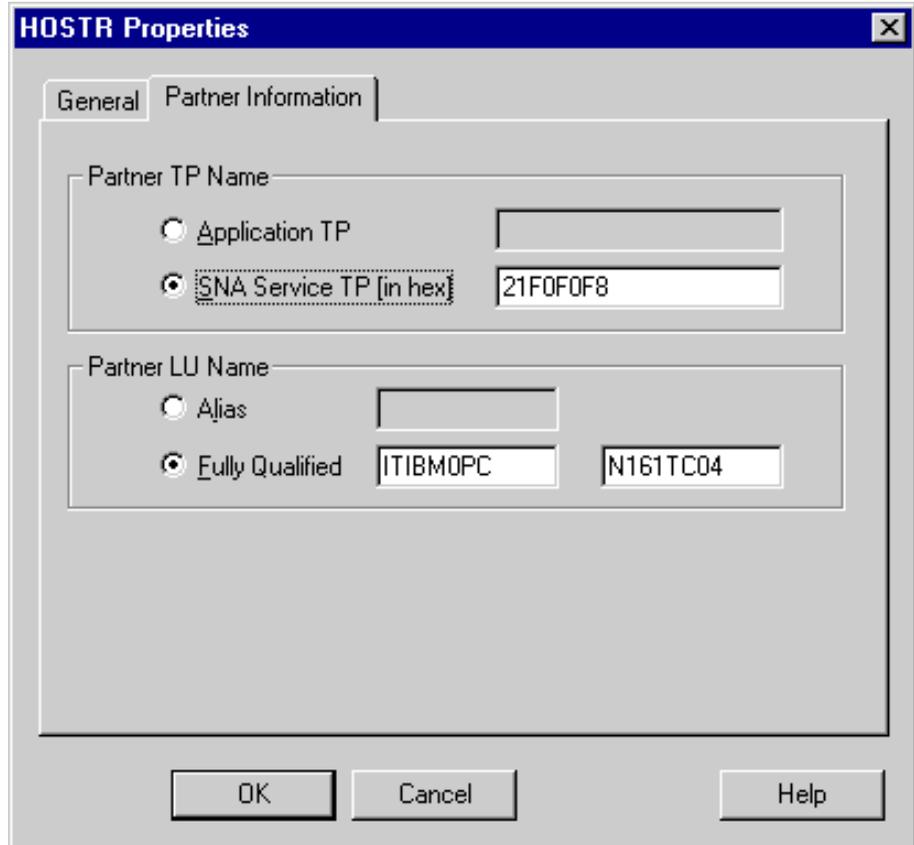


Figure 61. Configuring CPIC properties: partner information page for HOSTR

- 6 On the Partner Information page, enter the same values that are shown in Figure 61.
 - a. In the Partner TP Name group box, select the **SNA Service TP [in hex]** option button.
 - b. In the Partner LU Name group box, select the **Fully Qualified** option button.
- 7 Click **OK** to return to the Configuring CPIC Properties window.

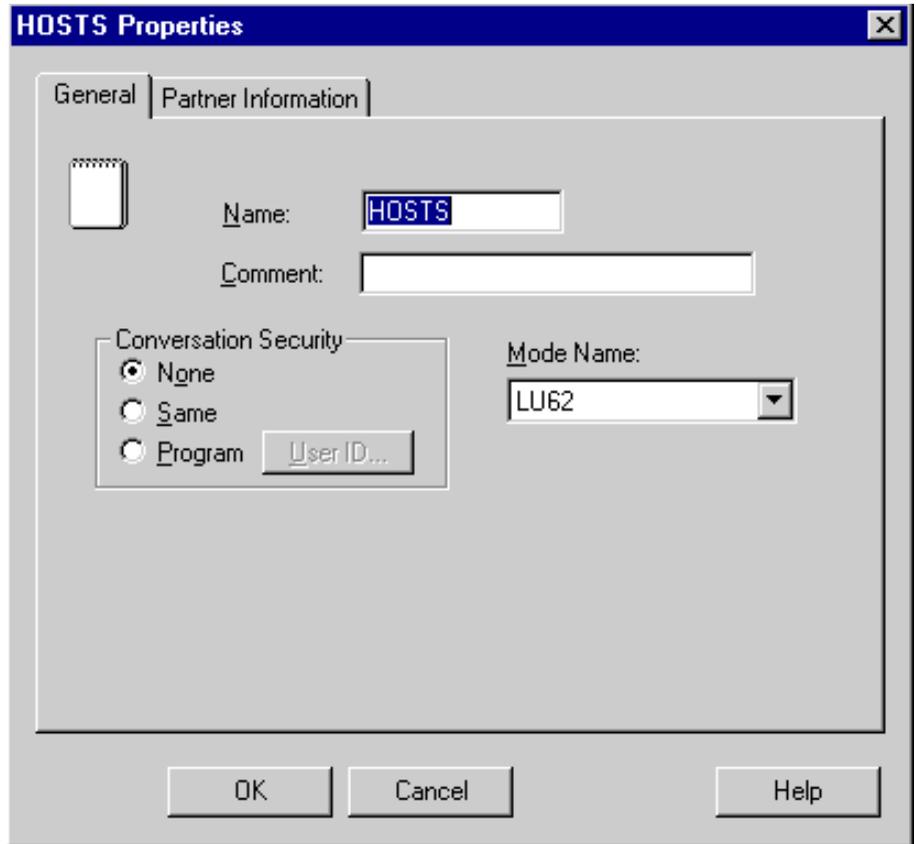


Figure 62. Configuring CPI-C properties: general page for HOSTS

- 8 On the General page, enter **HOSTS** in the Name field, as the CPI-C Symbolic Destination Name, as shown in Figure 62.
- 9 Click the **None** option button in the Conversation Security group box.
- 10 Select **LU62** in the Mode Name drop-down list.
- 11 Select the **Partner Information** tab. The Partner Information page appears.

Configuring the Local Transaction Program

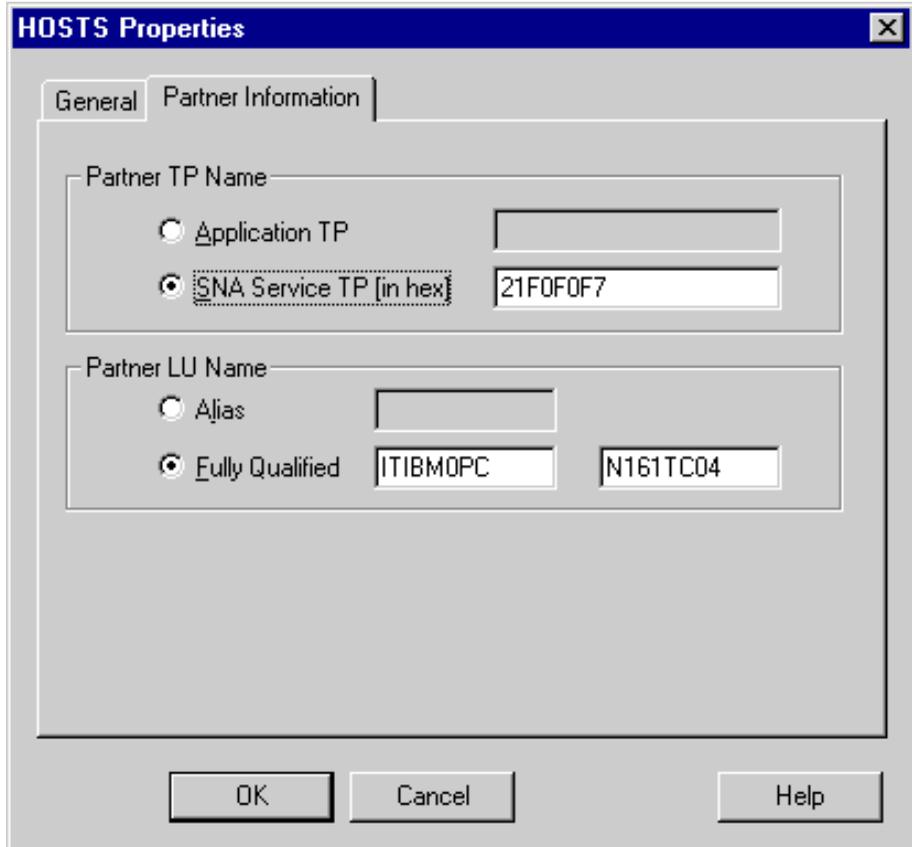


Figure 63. Configuring CPIC properties: partner information for HOSTS

- 12 On the Partner Information page, enter the same values that are shown in Figure 63.
 - a. In the Partner TP Name group box, select the **SNA Service TP [in hex]** option button.
 - b. In the Partner LU Name group box, select the **Fully Qualified** option button.

Configuring the Local Transaction Program

To configure the Local Transaction program for the send and receive functions, perform the following steps:

1. Go to the DOS prompt.
2. Change to the *productdir> \BIN* directory and run the following command:
FNDTPSET

The Invokable TP Setup window appears.

Configuring the Local Transaction Program for the Send Function

Figure 64. Configuring the local transaction program for the Send function

To configure the Local Transaction Program for the send function, perform the following steps:

1. Enter the values in the following fields as shown in Figure 64:
 - a. **SNAD007** in the **TP Name** field
 - b. **C:\SOFTDIST\BIN\FNDTS.EXE** in the **Command Line** field
 - c. **I9RLM000** in the **Local LU Alias** field
 - d. Deselect the **Set Up as Windows NT Service** check box
 - e. Deselect the **Queued Transaction Program** check box
2. Click **OK**.

Configuring the Local Transaction Program

Configuring the Local Transaction Program for the Retrieve Function

The screenshot shows the 'Invokable TP Setup' dialog box. The fields are filled with the following values:

- TP Name: SNAD008
- Command line: C:\SOFTDIST\BIN\FNDTR.EXE
- Local LU Alias: I9RLM000
- Timeout: Infinite (selected)

The 'Conversation Security User Accounts' section is currently empty.

Figure 65. Configuring the local transaction program for the Receive function

To configure the Local Transaction Program for the retrieve function, perform the following steps:

1. Enter values in the following fields as shown in Figure 65:
 - a. **SNAD008** in the **TP Name** field
 - b. **C:\SOFTDIST\BIN\FNDTR.EXE** in the **Command Line** field
 - c. **I9RLM000** in the **Local LU Alias** field
 - d. Deselect the **Set Up as Windows NT Service** check box
 - e. Deselect the **Queued Transaction Program** check box
2. Click **OK**.

Chapter 15. Configuring IBM Communications Server

This chapter explains how to configure the IBM Communications Server. You must configure IBM Communications Server to enable TME 10 Software Distribution, Version 3.1.5 for Windows NT to communicate with NetView DM for MVS.

The IBM Communication Server documentation contains additional information about how to configure the product.

Prerequisite

For this scenario the minimum version of Communications Server that you need to install depends on the operating system on which it will run:

Windows 2000 Communications Server, Version 6.1

Windows NT Communications Server, Version 5.0.2

General Assumptions

For this scenario it is assumed that:

- The workstation is connected with the host via Token Ring
- Local LU is LT0124A0
- CP Name is PPE0124
- Network name is ITIBM0PC
- Local Node ID is 05D-00124
- Partner LU is D3C73D01
- Partner CP name is MVSESA31
- Partner Network name is ITIBM0PC
- Configuration File name is PPE0124

Configuring IBM Communications Server Parameters

To configure the Communications Server parameters perform the following steps:

- 1** Select the **Start** push button then **Programs ► IBM Communications Server ► SNA - Node Configuration**. The Communications Server SNA Node Configuration window appears.

Configuring IBM Communications Server Parameters

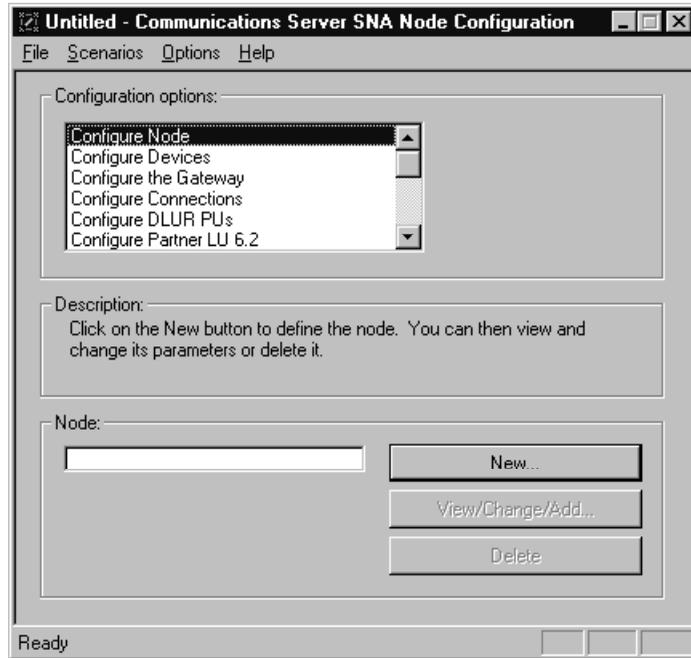


Figure 66. Communications Server SNA Node configuration window

- 2 From the pull-down menu select **File ▶New**.
- 3 Select **Scenarios ▶Advanced**.
- 4 From the Configuration options list box select **Configure Node** and then **New**, as shown in Figure 66. The Define the Node window appears.

Figure 67. Define the node window: basic page

Defining the Node

To define the node perform the following steps:

- 1 On the Basic page, in the Control Point (CP) box, use the values that are shown in Figure 67 for the following fields:
 - a. The fully qualified CP name. It is composed of the Network name plus the CP name.
 - b. The CP alias
- 2 In the Local Node ID box enter the values that are shown in Figure 67 for the following fields:
 - a. The block ID
 - b. The physical unit ID
- 3 In the Node Type list box select **End Node** as shown in Figure 67.
- 4 Select **Advanced**.

The Define the Node window showing the Advanced page appears.

Defining the Node

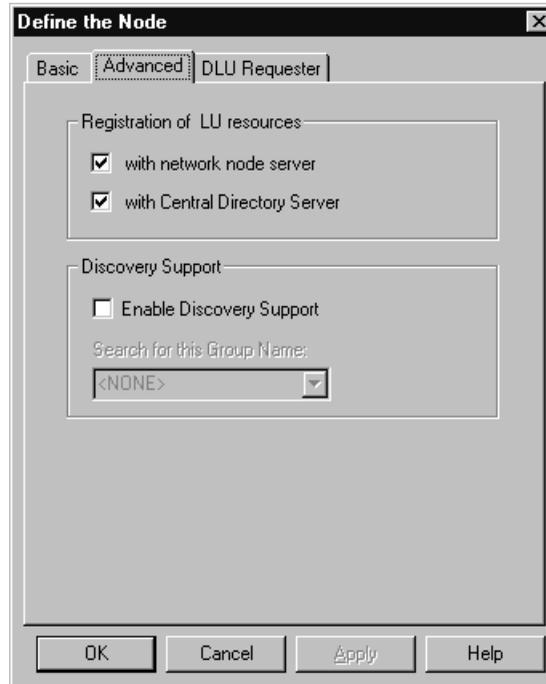


Figure 68. Define the node window: advanced page

- 5 On the Advanced page, in the Registration of LU resources box, select or deselect **with network node server** according to the operating system:

Windows 2000 Deselect **with network node server**

Windows NT Select **with network node server**

Select **with Central Directory Server**.

The example shown in Figure 68 is for Windows NT.

- 6 Select **DLU Requester**.

The Define the Node window showing the DLU Requester page appears.

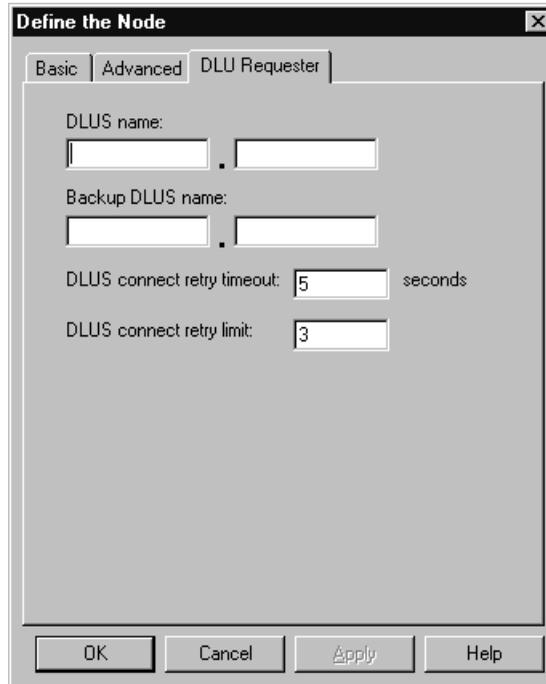


Figure 69. Define the node window: DLU requester page

- 7 For DLUS connect retry timeout and DLUS connect retry limit, use the same values that are shown in Figure 69.
- 8 Select **OK**.

Defining a LAN Device

To define a LAN device perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Devices**.
- 2 From the DLCs list box select **LAN** and then **New**.

The Define a LAN Device window showing the Basic page appears.

Defining a LAN Device

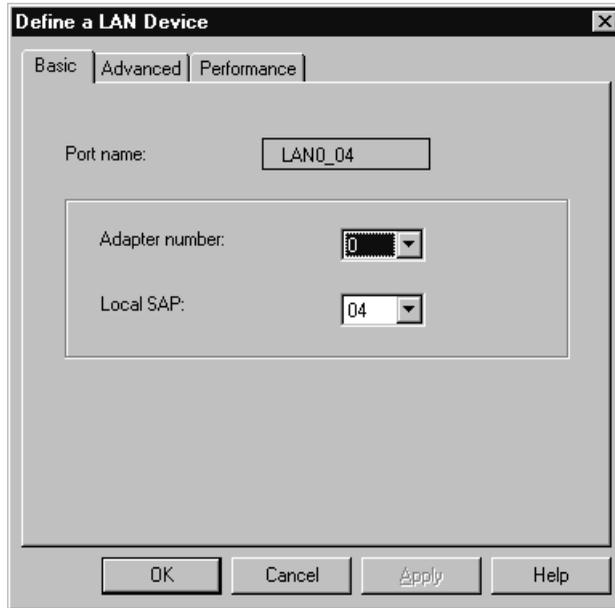


Figure 70. Define a LAN device window: basic page

- 3 On the Basic page, use the values that are shown in Figure 70 for the following fields:
 - a. Adapter number
 - b. Local SAP
- 4 Select **Advanced**.

The Define a LAN Device window showing the Advanced page appears.



Figure 71. Define a LAN device window: advanced page

5 On the Advanced page, use the same values that are shown in Figure 71.

6 Select **Performance**.

The Define a LAN Device window showing the Performance page appears.

Defining a LAN Connection

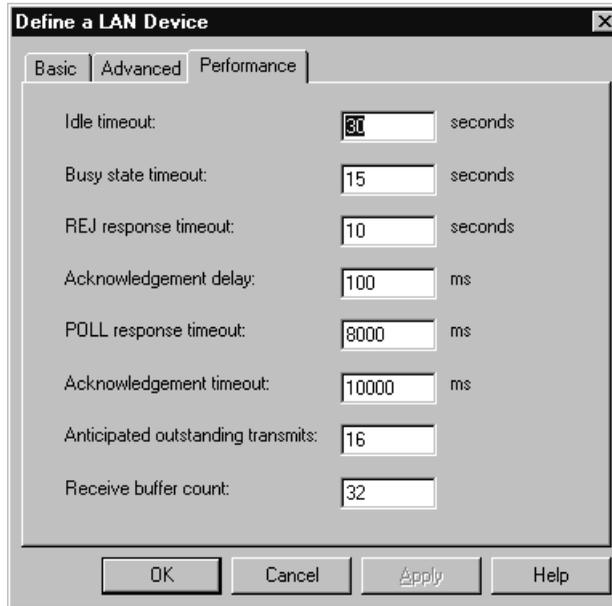


Figure 72. Define a LAN device window: performance page

- 7 On the Performance page, use the same values that are shown in the Figure 72.
- 8 Select **OK**.

Defining a LAN Connection

To define a LAN connection perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Connections**.
- 2 From the DLCs list box select **LAN** and then **New**.

The Define a LAN Connection window showing the Basic page appears.

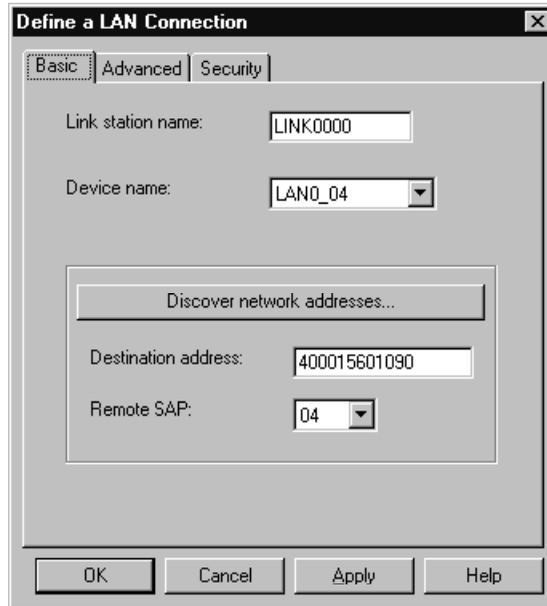


Figure 73. Define a LAN connection window: basic page

- 3 On the Basic page, use the values that are shown in Figure 73 for the following fields:
 - a. Enter the target link station name.
 - b. Select the device name.
 - c. Enter the destination address.
 - d. Select remote SAP.

4 Select Advanced.

The Define a LAN Connection Window Showing the Advanced page appears.

Defining a LAN Connection

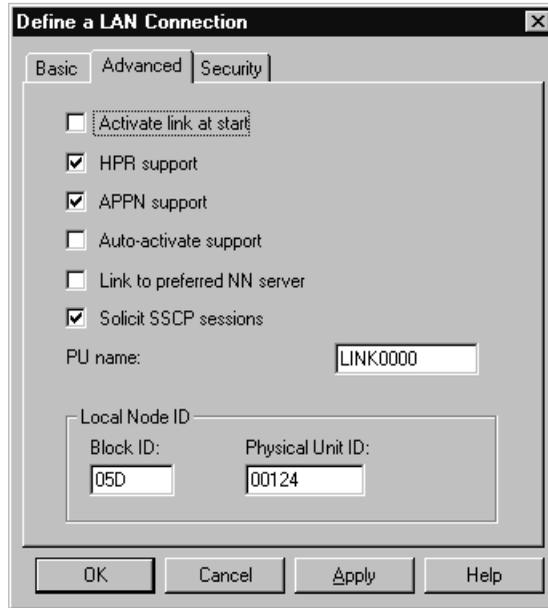


Figure 74. Define a LAN connection window: advanced page

5 On the Advanced page, use the same values that are shown in Figure 74.

6 Select **Security**.

The Define a LAN Connection window showing the Security page appears.

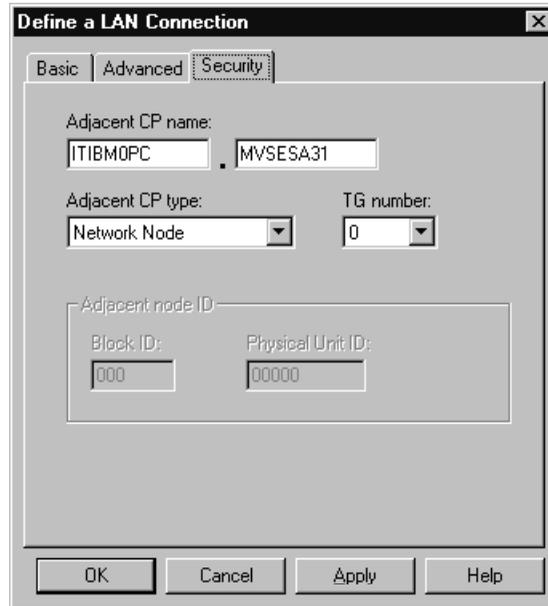


Figure 75. Define a LAN connection window: security page

- 7 On the Security page, use the same values that are shown in Figure 75 for the following fields:
 - a. The adjacent CP name
 - b. The adjacent CP type and the TG number
- 8 Select **OK**.

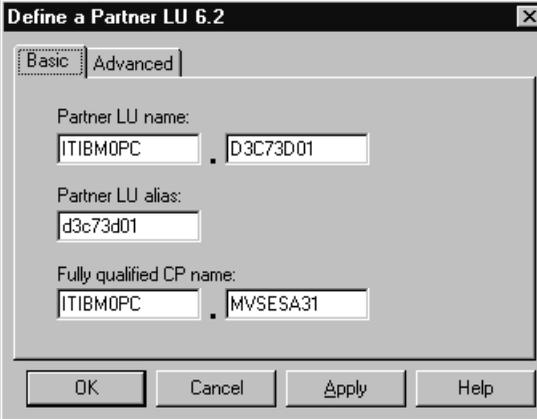
Defining a Partner LU 6.2

To define a partner LU 6.2 perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Partner LU 6.2** and then **New**.

The Define a Partner LU 6.2 window showing the Basic page appears.

Defining a Partner LU 6.2



The screenshot shows the 'Define a Partner LU 6.2' dialog box with the 'Basic' tab selected. The fields are filled with the following values:

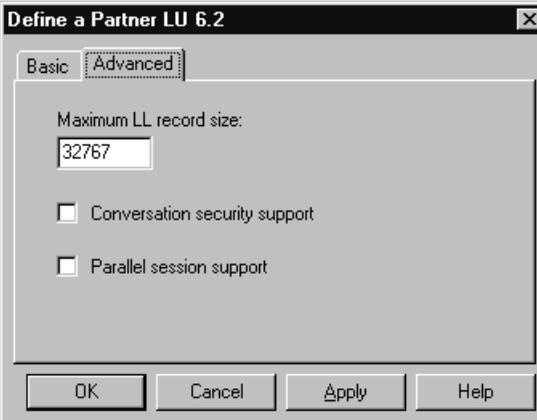
- Partner LU name: ITIBM0PC . D3C73D01
- Partner LU alias: d3c73d01
- Fully qualified CP name: ITIBM0PC . MVSESA31

Buttons at the bottom: OK, Cancel, Apply, Help.

Figure 76. Define a partner LU 6.2 window: basic page

- 2 On the Basic page, use the same values that are shown in Figure 76 for the following fields:
 - a. Partner LU name
 - b. Partner LU alias
 - c. Fully qualified CP name
- 3 Select **Advanced**.

The Define a Partner LU 6.2 window showing the Advanced page appears.



The screenshot shows the 'Define a Partner LU 6.2' dialog box with the 'Advanced' tab selected. The fields are filled with the following values:

- Maximum LL record size: 32767
- Conversation security support:
- Parallel session support:

Buttons at the bottom: OK, Cancel, Apply, Help.

Figure 77. Define a partner LU 6.2 window: advanced page

- 4 On the Advanced page, in the maximum LL record size field, enter the value that is shown in Figure 77. You do not select **Parallel session support** because the partner is a host.
- 5 Select **OK**.

Defining a Mode

To define a mode perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Mode** and then **New**.

The Define a Mode window showing the Basic page appears.

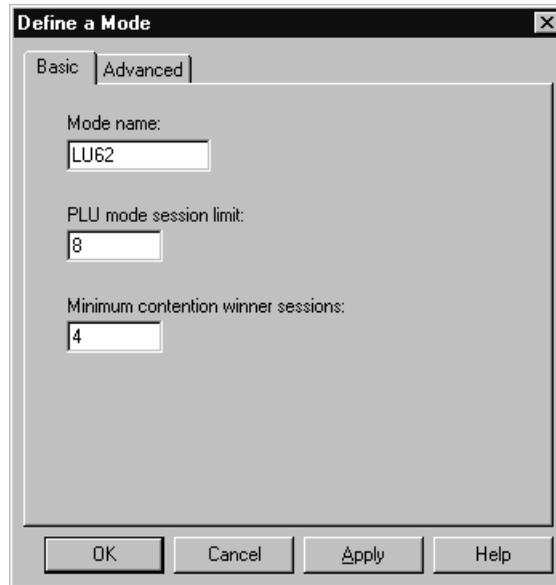


Figure 78. Define a mode window: basic page

- 2 On the Basic page, use the following values:

Mode name

LU62

PLU mode session limit

Windows 2000 32

Windows NT 8

Minimum contention winner sessions

Windows 2000 16

Windows NT 4

Figure 78 shows an example window for Windows NT.

- 3 Select **Advanced**.

The Define a Mode window showing the Advanced page appears.

Defining a Local LU 6.2

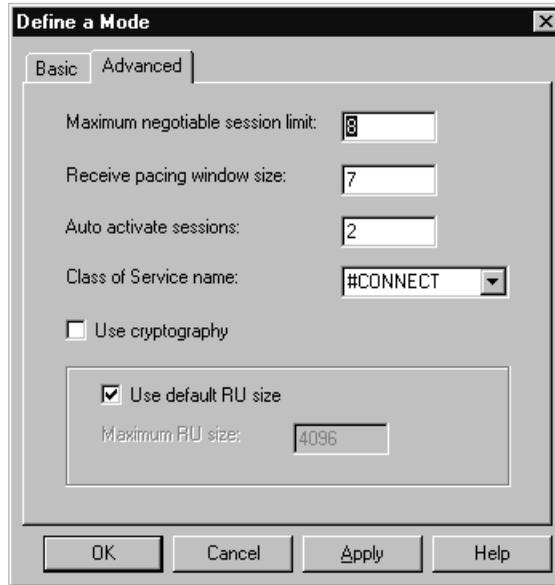


Figure 79. Define a mode window: advanced page

- 4 On the Advanced page perform the following steps:
 - In the entry fields enter the same values that are shown in Figure 79.
 - Select **CONNECT** for the Class of Service name.
 - Select the **Use default RU size** check box.
- 5 Select **OK**.

Defining a Local LU 6.2

To define a local LU 6.2 perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Local LU 6.2** and then **New**.
The Define a Local LU 6.2 window appears.



Figure 80. Define a local LU 6.2 window

- 2 Enter all the entry fields as shown in Figure 80.
- 3 Select **OK**.

Defining CPI-C Side Information for the Receive Function

You must define CPI-C side information for the receive function and for the send function.

This section shows how to define a CPI-C Side Information for the Receive Function. The “Defining CPI-C Side Information for the Send Function” on page 161 section shows how to define a transaction program for the send function. To define CPI-C side information for the receive function perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure CPI-C Side Information** and then **New**.

The Define CPI-C Side Information window showing the Basic page appears.

Defining CPI-C Side Information for the Receive Function

The screenshot shows a dialog box titled "Define CPI-C Side Information" with a close button (X) in the top right corner. It has two tabs: "Basic" and "Security". The "Basic" tab is selected. The dialog contains the following fields and options:

- Symbolic destination name:
- Mode name:
- Radio buttons: Use partner LU name, Use partner LU alias
- Partner LU name: .
- Partner LU alias:
- TP name:
- Check box: Service TP

At the bottom, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 81. Define CPI-C side information basic window for the receive function

- 2 On the Basic page, use the same values that are shown in Figure 81 for the following fields:
 - a. Enter **HOSTR** in the symbolic destination name field.

The symbolic destination name must match the receive **TP Symbolic Destination** field of the SNA/DS connection configuration file. The SNA/DS connection configuration file is stored in the <Target_dir>\DB\SNADSCON directory. See the Chapter 13, "Configuring SNA/DS Remote Connection Files" on page 105 for detailed information.
 - b. Select **LU62** in the Mode name field.
 - c. Select the **Use partner LU alias** radio button.
 - d. Select **d3c73d01** in Partner LU alias.
 - e. Enter **21008** in the TP name field.
 - f. Select the **Service TP** check box.
- 3 Select **Security**.

The Define CPI-C Side Information window showing the Security page appears.

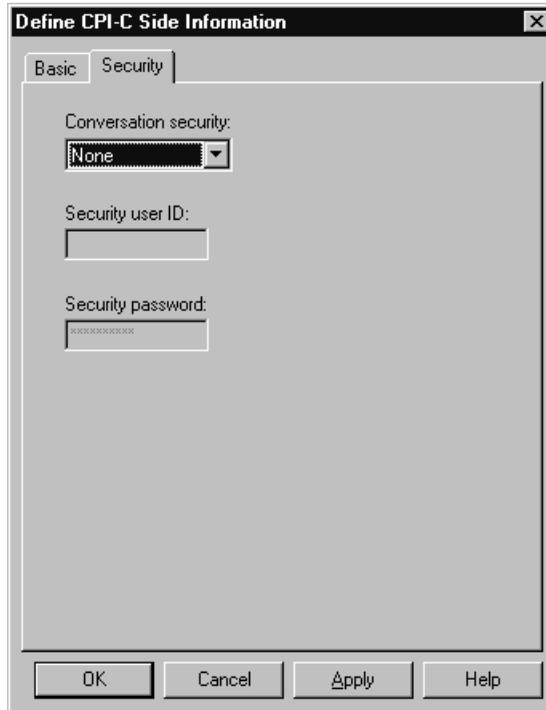


Figure 82. Define CPI-C side information window: security page

- 4 Select **none** for Conversation Security as shown in Figure 82.
- 5 Select **OK**.

Defining CPI-C Side Information for the Send Function

To define a CPI-C side information for the send function perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure CPI-C Side Information** and then **New**.

The Define CPI-C Side Information Basic window appears.

Defining CPI-C Side Information for the Send Function

The screenshot shows a dialog box titled "Define CPI-C Side Information" with a close button (X) in the top right corner. It has two tabs: "Basic" and "Security". The "Basic" tab is selected. The dialog contains the following fields and options:

- Symbolic destination name: A text box containing "HOSTS".
- Mode name: A dropdown menu showing "LU62".
- Radio buttons for "Use partner LU name" (unselected) and "Use partner LU alias" (selected).
- Partner LU name: Two empty text boxes separated by a period.
- Partner LU alias: A dropdown menu showing "d3c73d01".
- TP name: A text box containing "21007".
- A checked checkbox labeled "Service TP".

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 83. Define CPI-C side information basic window for the send function

2 On the Basic page, enter the following information as shown in Figure 83:

- a. Enter **HOSTS** in the Symbolic destination name field.

The symbolic destination name must match the send **TP Symbolic Destination** field of the SNA/DS connection configuration file. The SNA/DS connection configuration file is stored in the <Target_dir>\DB\SNADSCON directory. See Chapter 13, "Configuring SNA/DS Remote Connection Files" on page 105 for detailed information.

- b. Select **LU62** in the Mode name field.
c. Select the **Use partner LU alias** radio button.
d. Select **d3c73d01** in Partner LU alias.
e. Enter **21007** in the TP name field.
f. Select the **Service TP** check box.

3 Select **Security**.

The Define CPI-C Side Information window showing the Security page appears.



Figure 84. Define CPI-C side information window: security page

- 4 Select **none** for conversation security as shown in Figure 84.
- 5 Select **OK**.

Defining a Transaction Program for the Send Function

You must define transaction programs for the send function and for the receive function.

This section shows how to define a transaction program for the send function. The “Defining a Transaction Program for the Receive Function” on page 165 section shows how to define a transaction program for the receive function.

To define a transaction program for the send function perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Transaction Programs** and then **New**.

The Define a Transaction Program window showing the Basic page appears.

Defining a Transaction Program for the Send Function

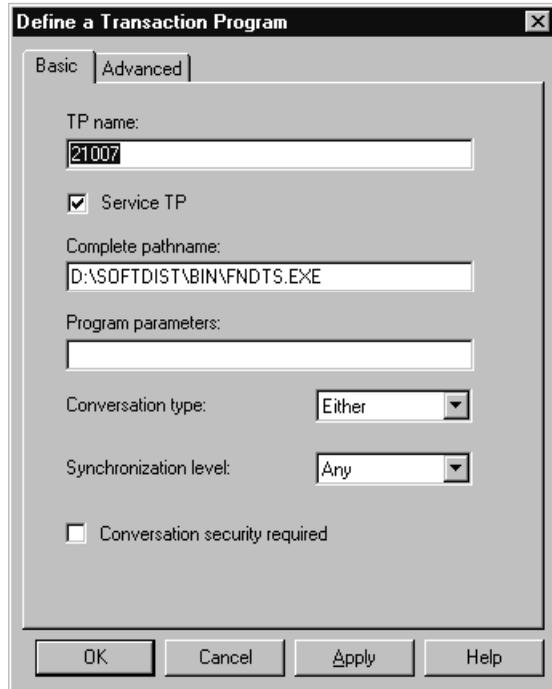


Figure 85. Define a transaction program basic window for the send function

On the Basic page use the same values that are shown in Figure 85 for the following fields:

- a. Enter **21007** in the TP name field.
- b. Select the Service TP check box.
- c. Enter **<Target_dir>\BIN\FNDTS.EXE** in the complete path name field.
- d. Select **Either** in the conversation type field.
- e. Select **Any** in the synchronization level field.

2 Select **Advanced**.

The Define a Transaction Program window showing the Advanced page appears.

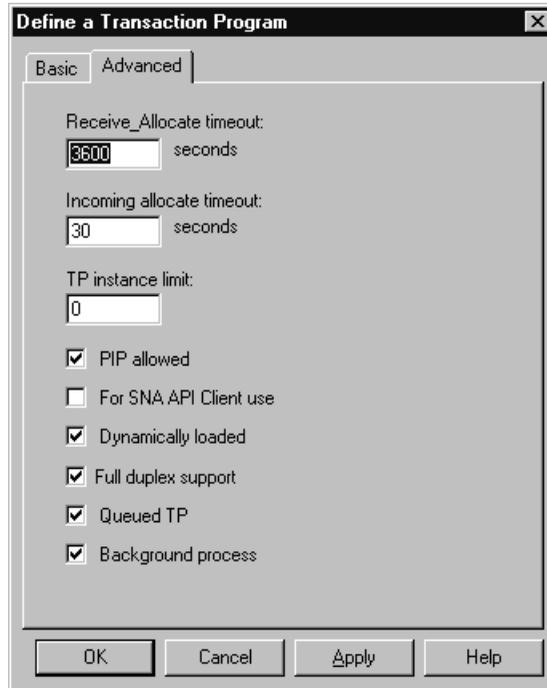


Figure 86. Define a transaction program window: advanced page

- 3 On the Advanced page do the following:
 - a. In the entry fields enter the values that are shown in Figure 86.
 - b. Select the following check boxes:
 - PIP allowed
 - Dynamically loaded
 - Full duplex support
 - Queued TP
 - Background process
- 4 Select **OK**.

Defining a Transaction Program for the Receive Function

To define a transaction program for the receive function perform the following steps:

- 1 From the Configuration options list box shown in Figure 66 on page 146 select **Configure Transaction Programs** and then **New**.

The Define a Transaction Program window showing the Basic page appears

Defining a Transaction Program for the Receive Function

The screenshot shows a dialog box titled "Define a Transaction Program" with a close button (X) in the top right corner. It has two tabs: "Basic" (selected) and "Advanced". The "Basic" tab contains the following fields and options:

- TP name:
- Service TP
- Complete pathname:
- Program parameters:
- Conversation type: (dropdown arrow)
- Synchronization level: (dropdown arrow)
- Conversation security required

At the bottom of the dialog box are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 87. Define CPI-C side Information basic window for the receive function

- 2 On the basic page enter the following information as shown in Figure 87:
 - a. Enter **21008** in the TP name field.
 - b. Select the **Service TP** check box.
 - c. Enter **D:\SOFTDIST\BIN\FNDTR.EXE** in the complete path name field.
 - d. Select **Either** in the conversation type field.
 - e. Select **Any** in the synchronization level field.

- 3 Select **Advanced**.

The Define a Transaction Program window showing the Advanced page appears.

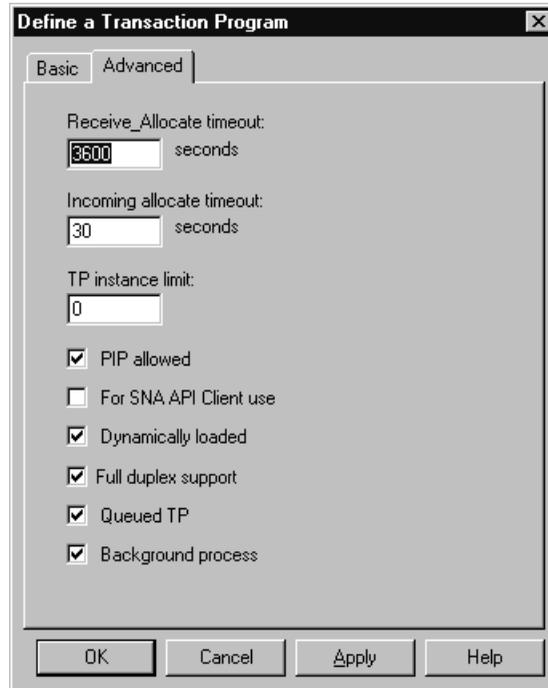


Figure 88. Define a transaction program window: advanced page

- 4 On the Advanced page perform the following steps:
 - a. In the entry fields enter the same values that are shown in Figure 88.
 - b. Select the following check boxes:
 - PIP allowed
 - Dynamically loaded
 - Full duplex support
 - Queued TP
 - Background process
- 5 Select **OK**.

Saving the Configuration File

After you have configured the configuration options perform the following steps to save them in a file:

- 1 Select **File ► Save as**.
- 2 Enter a configuration file name and then select **Exit**.

The file that contains the configuration is in the IBM Communications Server <target_dir>\private\<filename>.\acg directory.

Starting the Node

Starting the Node

After you perform the configuration steps do the following to start the node:

- 1 Select the **Start** push button then **Programs ► IBM Communications Server ► SNA Node Operations**. The Communications Server Node Operations window appears.
- 2 Select **Operations ► Start Node**. The Open window appears.

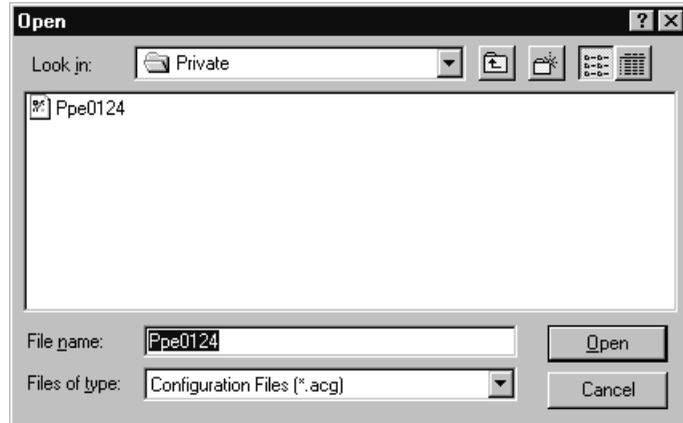


Figure 89. Open window

- 3 Select the configuration file that you created and then **Open**. The Communications Server Node Operations window containing the node characteristics appears.

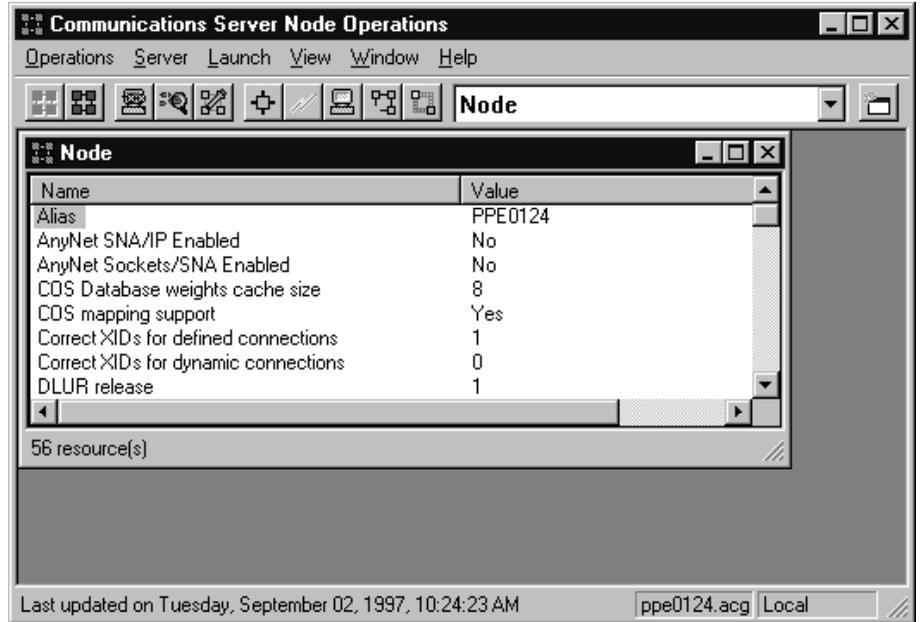


Figure 90. Node characteristics window

- 4 Select **Connections** from the list box.

Starting the Node

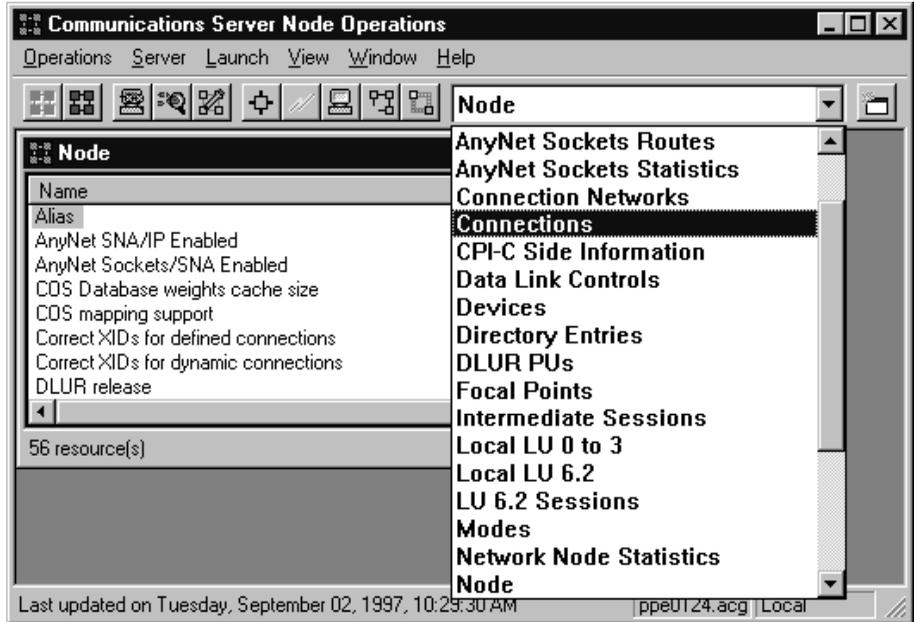


Figure 91. Node characteristics connection selection window

The Communications Server Node Operations Connections window appears.

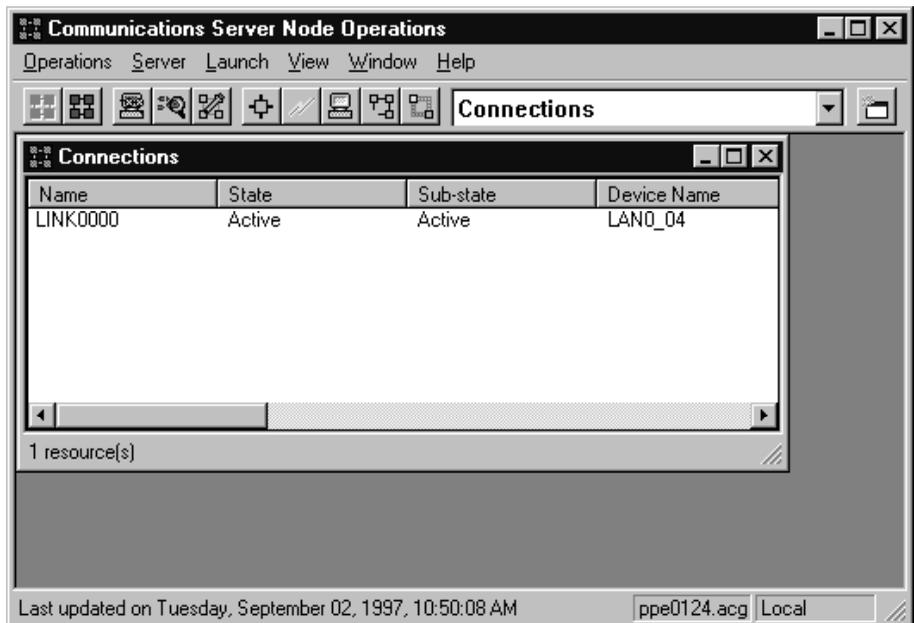


Figure 92. Node characteristics connection window

- 5 Select the name, and press the right mouse button. A pull down menu appears.
- 6 Select **Start**, than press the left mouse button or select **Enter**.

Establishing an APPC Connection between the TME 10 Software Distribution for Windows NT Server and Another TME 10 Software Distribution, Version 3.1.5 Server

To establish an APPC connection between the TME 10 Software Distribution for Windows NT server and another TME 10 Software Distribution, Version 3.1.5 server perform this scenario with the following exceptions:

- Enter the correct information for the remote target and select the **Parallel section support** check box that is shown in Figure 77 on page 156.
- Before you establish the APPC connection, set APPCLLU=<NT Local LU Name> as system environment variable in the Windows NT client. In this scenario APPCLLU assumes the following values:

```
APPCLLU=LT0124A0
```

You must set the APPCLLU system variable because the CP name of the Windows NT workstation is used as default logical unit name by the Communications Server.

Defining the Windows NT Node at NetView DM for MVS Using GIX

To define the Windows NT node at NetView DM for MVS you do one of the following:

- Specify the CP name on the Windows NT workstation in the Logical Unit field of the NODE ATTRIBUTES panel in the GIX environment.
- Set APPCLLU=<NT Local LU Name> as system environment variable in the Windows NT client. In this scenario APPCLLU assumes the following values:

```
APPCLLU=LT0124A0
```

You have to specify the CP name in the Logical Unit Field because the CP name of the Windows NT workstation is used as default logical unit name by the Communications Server.

Chapter 16. Configuring IBM Personal Communications

This chapter explains how to configure IBM Personal Communications. You must configure Personal Communications to enable TME 10 Software Distribution, Version 3.1.5 for Windows NT to communicate with NetView DM for MVS.

The Personal Communications documentation contains additional information about how to configure the product.

Prerequisite

For this scenario it is assumed you will install APAR IC18696 in addition to Personal Communications, Version 4.1 or later.

General Assumptions

For this scenario it is assumed that:

- The workstation is connected with the host via Token Ring
- Local LU is LT0071A0
- CP Name is PPE0071
- Network name is ITIBM0PC
- Local Node ID is 05D-00071
- Partner LU is D3C73D01
- Partner CP name is MVSESA31
- Partner Network name is ITIBM0PC
- Configuration File name is PPE0071

Configuring IBM Personal Communications Parameters

To configure the Personal Communications parameters perform the following steps:

- 1 Select the **Start** push button then **Programs ► IBM Personal Communications ► SNA - Node Configuration**. The Personal Communications SNA Node Configuration window appears.

Configuring IBM Personal Communications

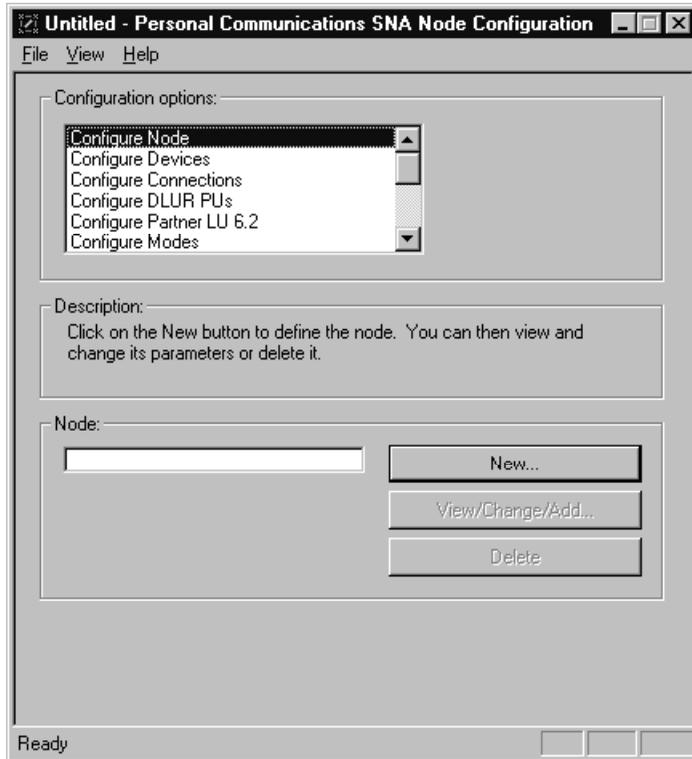


Figure 93. Personal Communications SNA node configuration window

- 2 From the Configuration options list box select **Configure Node** and then **New**, as shown in Figure 93. The Define the Node window appears.

Figure 94. Define the node: basic page

Defining the Node

To define the node perform the following steps:

- 1 On the Basic page, in the Control Point (CP) box, use the values that are shown in Figure 94 for the following fields:
 - a. The fully qualified CP name. It is composed of the Network name plus the CP name.
 - b. The CP alias
- 2 In the Local Node ID box enter the values that are shown in Figure 94 for the following fields:
 - a. The block ID
 - b. The physical unit ID
- 3 Select **Advanced**.

The Define the Node window showing the Advanced page appears.

Defining the Node

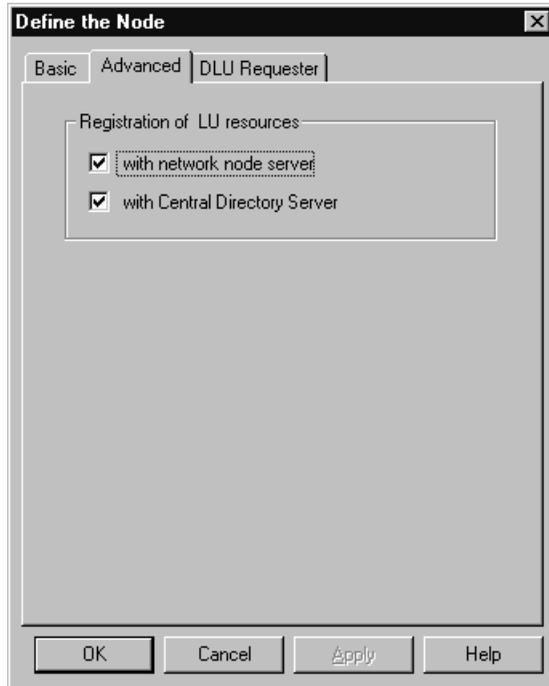


Figure 95. Define the node: advanced page

4 On the Advanced page, in the Registration of LU resources box, select **with network node server** and **with Central Directory Server** as shown in Figure 95.

5 Select **DLU Requester**.

The Define the Node window showing the DLU Requester page.

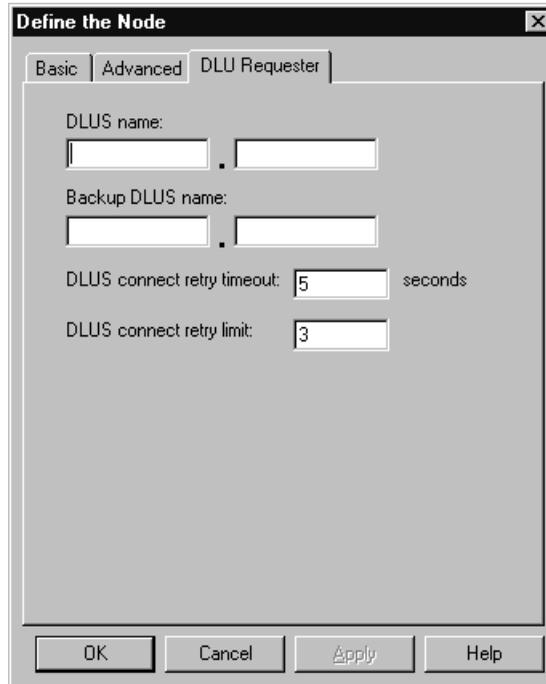


Figure 96. Define the node: DLU requester page

- 6 For DLUS connect retry timeout and DLUS connect retry limit, use the same values that are shown in Figure 96.
- 7 Select **OK**.

Defining a LAN Device

To define a LAN device perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure Devices**.
- 2 From the DLCs list box select **LAN** and then **New**.

The Define a LAN Device window showing the Basic page appears.

Defining a LAN Device

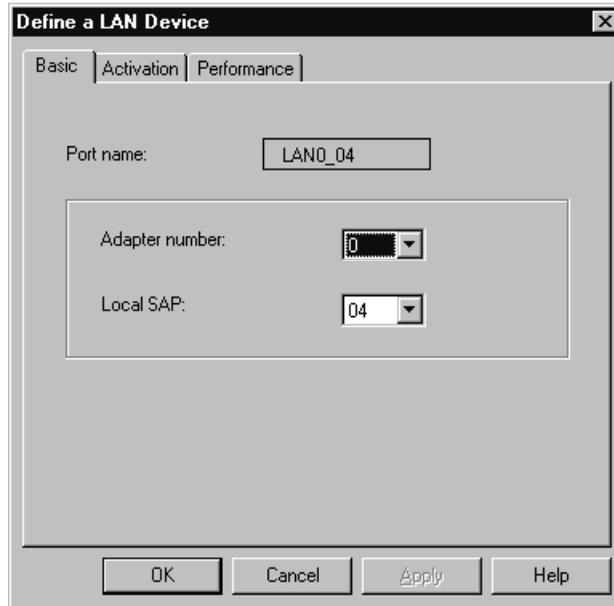


Figure 97. Define a LAN device: basic page

- 3 On the Basic page, use the values that are shown in Figure 97 for the following fields:
 - a. Adapter number
 - b. Local SAP
- 4 Select **Activation**.

The Define a LAN Device window showing the Activation page appears.

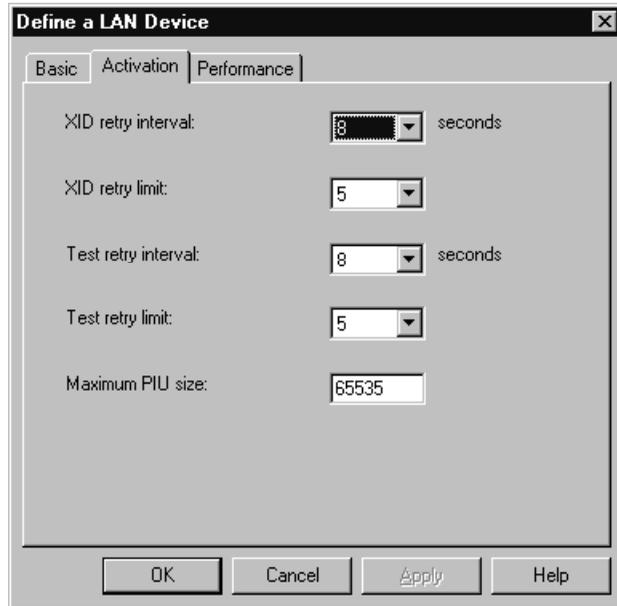


Figure 98. Define a LAN device: activation page

5 On the Activation page, use the same values that are shown in Figure 98.

6 Select **Performance**.

The Define a LAN Device window showing the Performance page appears.

Defining a LAN Connection

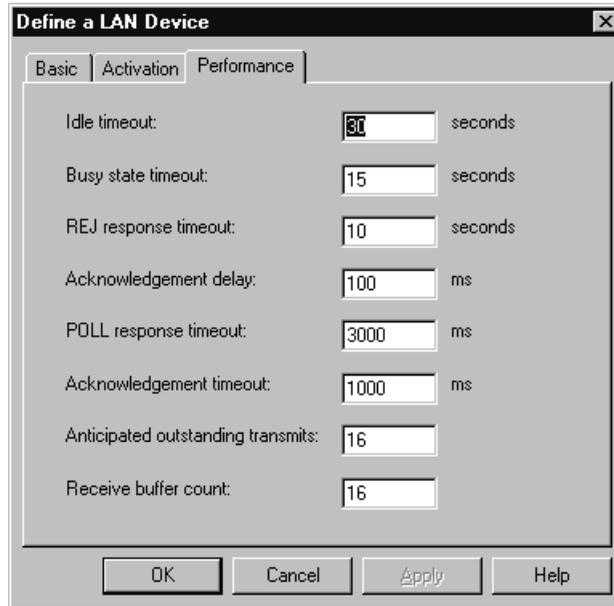


Figure 99. Define a LAN device: performance page

- 7 On the Performance page, use the same values that are shown in the Figure 99.
- 8 Select **OK**.

Defining a LAN Connection

To define a LAN connection perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure Connections**.
- 2 From the DLCs list box select **LAN** and then **New**.

The Define a LAN Connection window showing the Basic page appears.

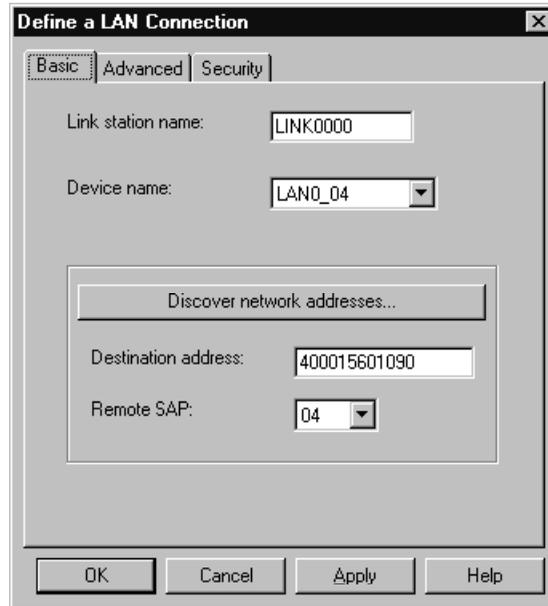


Figure 100. Define a LAN connection: basic page

- 3 On the Basic page, use the values that are shown in Figure 100 for the following fields:
 - a. Enter the target link station name.
 - b. Select the device name.
 - c. Enter the destination address.
 - d. Select remote SAP.

4 Select **Advanced**.

The Define a LAN Connection window showing the Advanced page appears.

Defining a LAN Connection

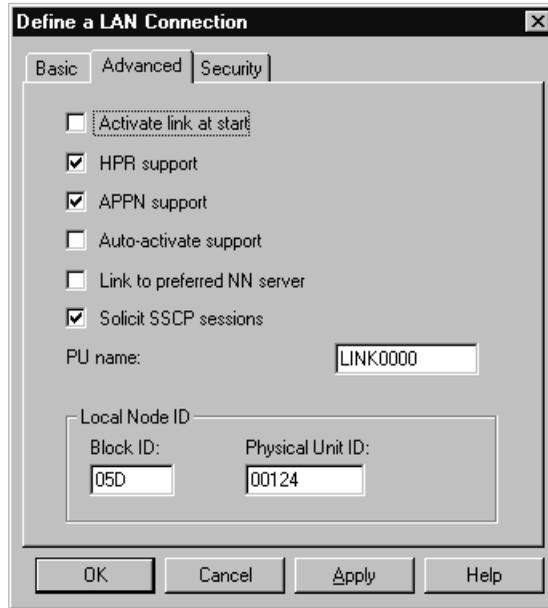


Figure 101. Define a LAN connection: advanced page

5 On the Advanced page, use the same values that are shown in Figure 101.

6 Select **Security**.

The Define a LAN Connection window showing the Security page appears.

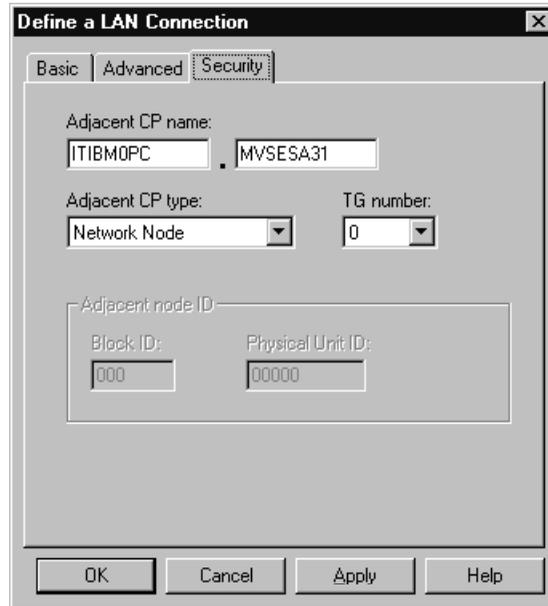


Figure 102. Define a LAN connection: security page

- 7 On the Security page, use the same values that are shown in Figure 102 for the following fields:
 - a. The adjacent CP name
 - b. The adjacent CP type and the TG number
- 8 Select **OK**.

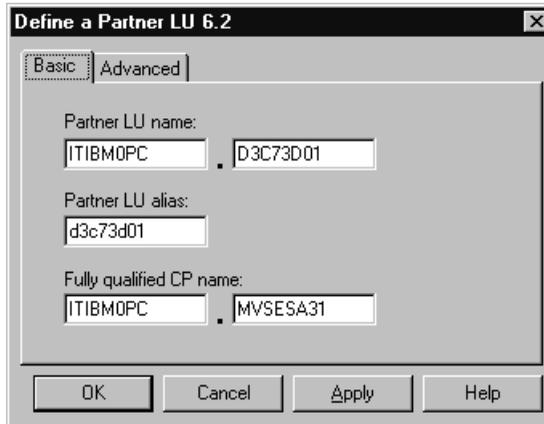
Defining a Partner LU 6.2

To define a partner LU 6.2 perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure Partner LU 6.2** and then **New**.

The Define a Partner LU 6.2 window showing the Basic page appears.

Defining a Partner LU 6.2



The screenshot shows the 'Define a Partner LU 6.2' dialog box with the 'Basic' tab selected. The fields are filled with the following values:

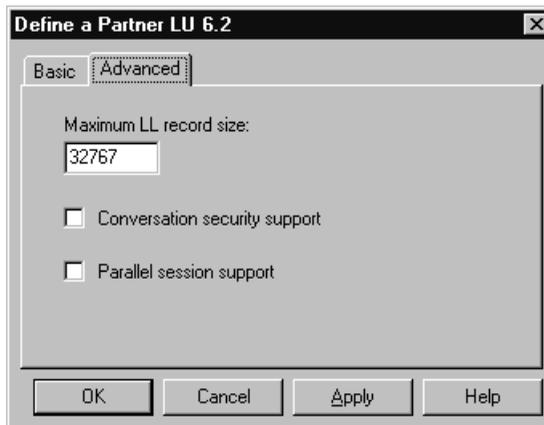
- Partner LU name: ITIBM0PC . D3C73D01
- Partner LU alias: d3c73d01
- Fully qualified CP name: ITIBM0PC . MVSESA31

Buttons at the bottom: OK, Cancel, Apply, Help.

Figure 103. Define a partner LU 6.2: basic page

- 2 On the Basic page, use the same values that are shown in Figure 103 for the following fields:
 - a. Partner LU name
 - b. Partner LU alias
 - c. Fully qualified CP name
- 3 Select **Advanced**.

The Define a Partner LU 6.2 window showing the Advanced page appears.



The screenshot shows the 'Define a Partner LU 6.2' dialog box with the 'Advanced' tab selected. The fields are filled with the following values:

- Maximum LL record size: 32767
- Conversation security support:
- Parallel session support:

Buttons at the bottom: OK, Cancel, Apply, Help.

Figure 104. Define a partner LU 6.2: advanced page

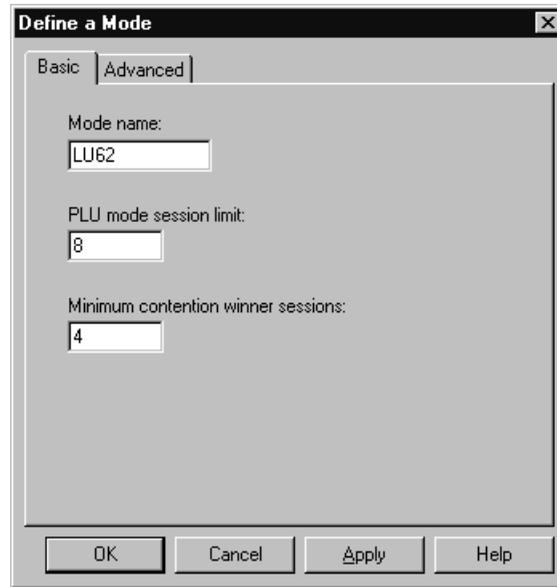
- 4 On the Advanced page, in the maximum LL record size field, enter the value that is shown in Figure 104. You do not select **Parallel session support** because the partner is a host.
- 5 Select **OK**.

Defining a Mode

To define a mode perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure Mode** and then **New**.

The Define a Mode window showing the Basic page appears.



The image shows a dialog box titled "Define a Mode" with a close button (X) in the top right corner. It has two tabs: "Basic" and "Advanced". The "Basic" tab is selected. Inside the dialog, there are three text input fields. The first is labeled "Mode name:" and contains the text "LU62". The second is labeled "PLU mode session limit:" and contains the number "8". The third is labeled "Minimum contention winner sessions:" and contains the number "4". At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 105. Define a mode: basic page

- 2 On the Basic page, use the same values that are shown in Figure 105 for the following fields:
 - a. Mode name
 - b. PLU mode session limit
 - c. Minimum contention winner sessions
- 3 Select **Advanced**.

The Define a Mode window showing the Advanced page appears.

Defining a Local LU 6.2

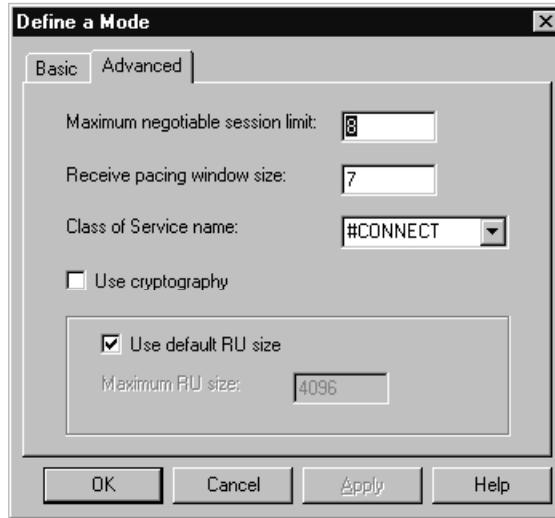


Figure 106. Define a mode: advanced page

- 4 On the Advanced page perform the following steps:
 - In the entry fields enter the same values that are shown in Figure 106.
 - Select **CONNECT** for the Class of Service name.
 - Select the **Use default RU size** check box.
- 5 Select **OK**.

Defining a Local LU 6.2

To define a local LU 6.2 perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure Local LU 6.2** and then **New**.
The Define a Local LU 6.2 window appears.

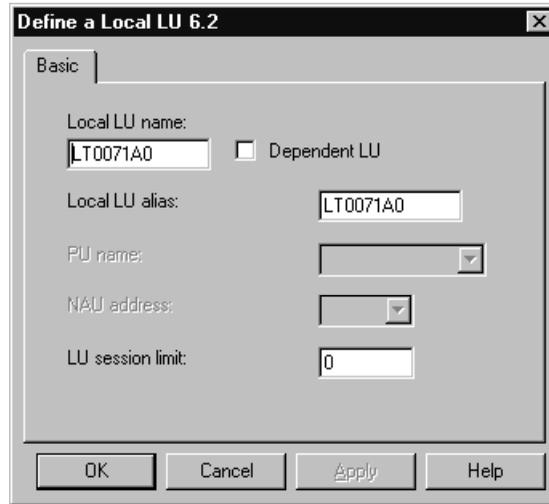


Figure 107. Define a local LU 6.2

- 2 Enter all the entry fields as shown in Figure 107.
- 3 Select **OK**.

Defining CPI-C Side Information for the Receive Function

You must define CPI-C side information for the receive function and for the send function.

This section shows how to define a CPI-C Side Information for the Receive Function. The “Defining CPI-C Side Information for the Send Function” on page 189 section shows how to define a transaction program for the send function. To define CPI-C side information for the receive function perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure CPI-C Side Information** and then **New**.

The Define CPI-C Side Information window showing the Basic page appears.

Defining CPI-C Side Information for the Receive Function

The screenshot shows a dialog box titled "Define CPI-C Side Information" with a close button (X) in the top right corner. It has two tabs: "Basic" and "Security". The "Basic" tab is active. The fields are as follows:

- Symbolic destination name:
- Mode name: (dropdown menu)
- Partner LU name: .
- TP name:
- Service TP

At the bottom, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Figure 108. Define CPI-C side information: basic page, for the RECEIVE function

- 2 On the Basic page, use the same values that are shown in Figure 108 for the following fields:

- a. Enter **HOSTR** in the symbolic destination name field.

The symbolic destination name must match the receive **TP Symbolic Destination** field of the SNA/DS connection configuration file. The SNA/DS connection configuration file is stored in the <Target_dir>\DB\SNADSCON directory. See Chapter 13, "Configuring SNA/DS Remote Connection Files" on page 105 for detailed information.

- b. Select **LU62** in Mode name.
- c. Select the **Use partner LU alias** radio button.
- d. Enter **ITIBM0PC.D3C73D01** in Partner LU name.
- e. Enter **21008** in the TP name field.
- f. Select the **Service TP** check box.

- 3 Select **Security**.

The Define CPI-C Side Information window showing the Security page appears.



Figure 109. Define CPI-C side information: security page

- 4 Select **none** for Conversation Security as shown in Figure 109.
- 5 Select **OK**.

Defining CPI-C Side Information for the Send Function

To define a CPI-C side information for the send function perform the following steps:

- 1 From the Configuration options list box shown in Figure 93 on page 174 select **Configure CPI-C Side Information** and then **New**.

The Define CPI-C Side Information showing the Basic page appears.

Defining CPI-C Side Information for the Send Function

The screenshot shows a dialog box titled "Define CPI-C Side Information" with a close button (X) in the top right corner. It has two tabs: "Basic" and "Security". The "Basic" tab is selected. The dialog contains the following fields and controls:

- Symbolic destination name: A text box containing "HOSTS".
- Mode name: A dropdown menu showing "LU62".
- Partner LU name: Two text boxes containing "ITIBM0PC" and ".D3C73D01".
- TP name: A text box containing "21007".
- Service TP: A checked checkbox.
- Buttons: "OK", "Cancel", "Apply", and "Help" at the bottom.

Figure 110. Define CPI-C side information: basic page, for the SEND function

2 On the Basic page, enter the following information as shown in Figure 110:

- a. Enter **HOSTS** in the Symbolic destination name field.

The symbolic destination name must match the send **TP Symbolic Destination** field of the SNA/DS connection configuration file. The SNA/DS connection configuration file is stored in the <Target_dir>\DB\SNADSCON directory. See Chapter 13, "Configuring SNA/DS Remote Connection Files" on page 105 for detailed information.

- b. Select **LU62** in the Mode name list box.
- c. Select the **Use partner LU alias** radio button.
- d. Enter **ITIBM0PC.D3C73D01** in Partner LU name field.
- e. Enter **21007** in the TP name field.
- f. Select the **Service TP** check box.

3 Select **Security**.

The Define CPI-C Side Information window showing the Security page appears.



Figure 111. Define CPI-C side information: security page

- 4 Select **none** for conversation security as shown in Figure 111.
- 5 Select **OK**.

Saving the Configuration File

After you have configured the configuration options perform the following steps to save them in a file:

- 1 Select **File ► Save as**.
- 2 Enter a configuration file name and then select **Exit**.

The file that contains the configuration is in the IBM Personal Communications <target_dir>\private\<filename>.pcg directory.

Configuring the Local Transaction Program

To configure the Local Transaction program you must use the FNDTPSET utility provided with the TME 10 Software Distribution, Version 3.1.5 for Windows NT product. To configure the Local Transaction program for the send and receive functions perform the following steps:

Configuring the Local Transaction Program

1. Select a DOS prompt.
2. Run the following command from the <productdir>\BIN directory:
FNDTPSET

The Invokable TP Setup window appears.

The screenshot shows the 'Invokable TP Setup' dialog box. The fields are filled as follows: TP Name: SNAD007; Command line: C:\SOFTDIST\BIN\FNDTS.EXE; Local LU Alias: LT0071A0. The 'Timeout' is set to 'Finite' with a value of 5 seconds. The 'Infinite' option is also present but not selected. The 'Conversation Security User Accounts' section is empty. The 'Set Up as Windows NT Service', 'Queued Transaction Program', 'Conversation Security', and 'Accepts Already Verified Passwords' checkboxes are all unchecked.

Figure 112. Configuring the local Transaction Program, for the SEND function

Configuring the Local Transaction Program for the Send Function

To configure the Local Transaction Program for the send function perform the following steps:

1. Fill in the following fields as shown in Figure 112:
 - a. The SNAD007 value in the **TP Name** field
 - b. The C:\SOFTDIST\BIN\FNDTS.EXE command in the **Command Line** field
 - c. The LT0071A0 value in the **Local LU Alias** field
 - d. Deselect the Set Up as Windows NT Service check box
 - e. Deselect the Queued Transaction Program check box
2. Select **OK**.

Configuring the Local Transaction Program for the Retrieve Function

To configure the Local Transaction Program for the retrieve function perform the following steps:

1. Fill in the following fields as shown in Figure 113 on page 193:

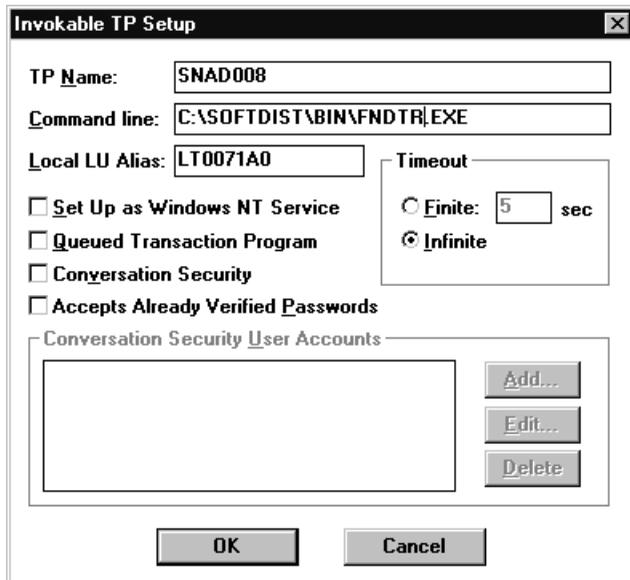


Figure 113. Configuring the local Transaction Program, for the RECEIVE function

- a. The SNAD008 value in the **TP Name** field
 - b. The C:\SOFTDIST\BIN\FNDTR.EXE command in the **Command Line** field
 - c. The LT0071A0 value in the **Local LU Alias** field
 - d. Deselect the Set Up as Windows NT Service check box
 - e. Deselect the Queued Transaction Program check box
2. Select **OK**.

Starting the Node

After you perform the configuration steps do the following to start the node:

- 1 Select the **Start** push button then **Programs ► IBM Personal Communications ► Administrative and PD Aids ► SNA Node Operations**. The Personal Communications SNA Node Operations window appears.
- 2 Select **Operations ► Start Node**. The Open window appears.

Starting the Node

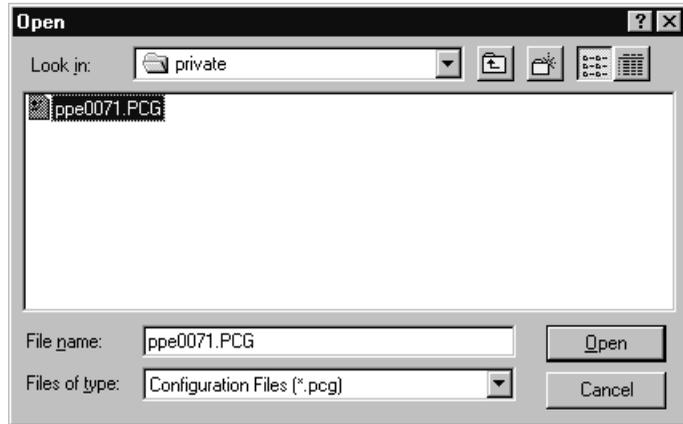


Figure 114. Open

- 3 Select the configuration file that you created and then **Open**. The Personal Communications SNA Node Operations: Nodes characteristics window appears.

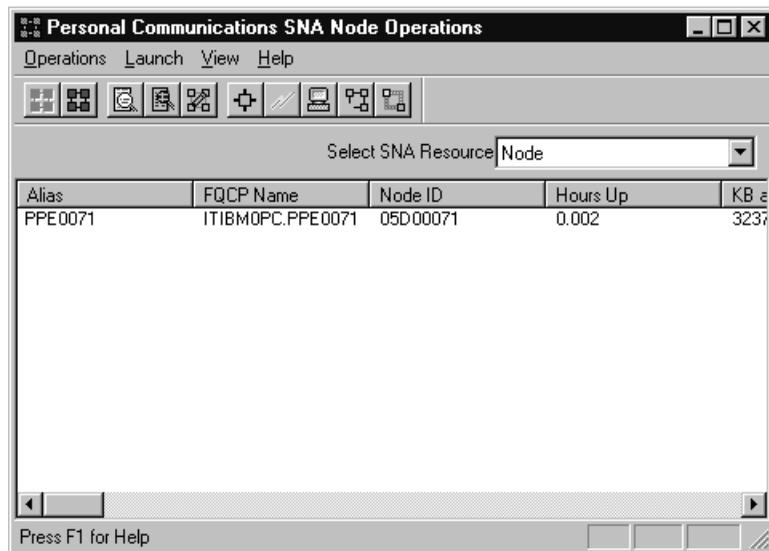


Figure 115. Node characteristics

- 4 Select **Connections** from the Select SNA Resource list box as shown in Figure 116 on page 195.

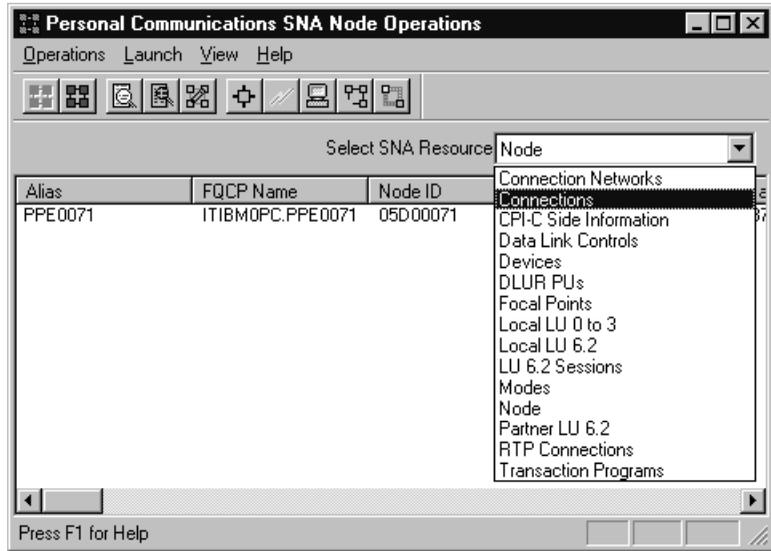


Figure 116. Node characteristics connections selection

The Personal Communications SNA Node Operations: Connections window appears.

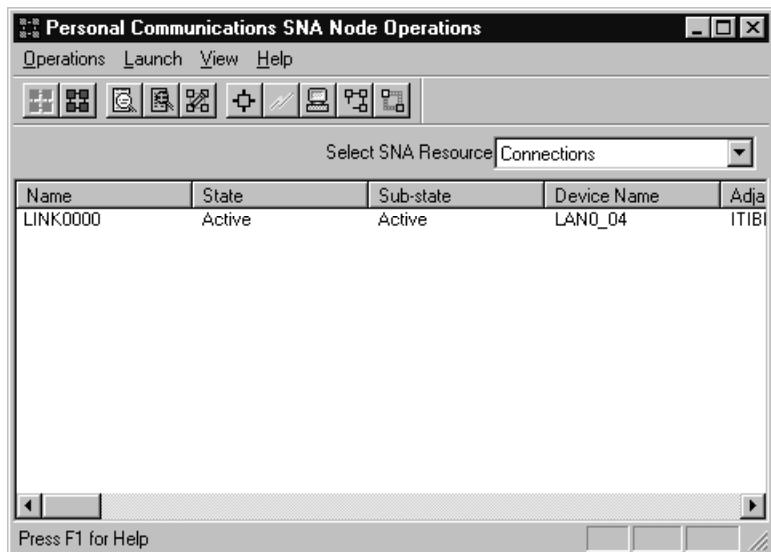


Figure 117. Node characteristics connections

- 5 Select the name, and press the right mouse button. A pull down menu appears.
- 6 Select **Start**, then press the left mouse button or select **Enter**.

Establishing an APPC Connection between the TME 10 Software Distribution for Windows NT Server and Another TME 10 Software Distribution, Version 3.1.5 Server

To establish an APPC connection between the TME 10 Software Distribution for Windows NT server and another TME 10 Software Distribution, Version 3.1.5 server, perform this scenario with the following exceptions:

- Enter the correct information for the remote target and select the **Parallel section support** check box that is shown in Figure 104 on page 184.
- Before you establish an APPC connection, set APPCLLU=<NT Local LU Name> as system environment variable. In this scenario APPCLLU assumes the following values:

```
APPCLLU=LT0071A0
```

You must set the APPCLLU system variable because the CP name of the Windows NT workstation is used as default logical unit name by the Communications Server.

Defining the Windows NT Node at NetView DM for MVS Using GIX

To define the Windows NT node at NetView DM for MVS using GIX, you do one of the following:

- Specify the CP name on the Windows NT workstation in the Logical Unit field of the NODE ATTRIBUTES panel in the GIX environment.
- Set APPCLLU=<NT Local LU Name> as system environment variable in the Windows NT client. In this scenario APPCLLU assumes the following values:

```
APPCLLU=LT0071A0
```

You have to specify the CP name in the Logical Unit Field because the CP name of the Windows NT workstation is used as the default logical unit name by IBM Personal Communications.

Using the Personal Communications GUIs with TME 10 Software Distribution

To use the Personal Communications GUIs on the same workstation where TME 10 Software Distribution is installed, you must start at least one Personal Communications GUI (for example the Personal Communications SNA Node Operation GUI) before you start TME 10 Software Distribution. This means that you must start TME 10 Software Distribution manually. To configure TME 10 Software Distribution to be started manually perform the following steps:

- 1** Select the **Start** push button then **Settings ► Control Panel**. The Control Panel window appears.
- 2** Select the **Services** icon. The Services window appears.
- 3** Select **NetView DM** from the Service list and then the **Startup...** push button. The Service window appears.

| **4** Select the **Manual** radio button and then **OK**.

| **5** At the next Windows NT operating system start up open a DOS Command
| Prompt and enter the following command:

| NET START NETVIEWDM

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
- Prepare software using DiskCamera

Chapter 17. Software Distribution Scenarios

The following scenarios show how to prepare applications for distribution.

The Software Distribution Environment

Before running the scenarios, be sure that you have, for example, the environment shown in Figure 118.

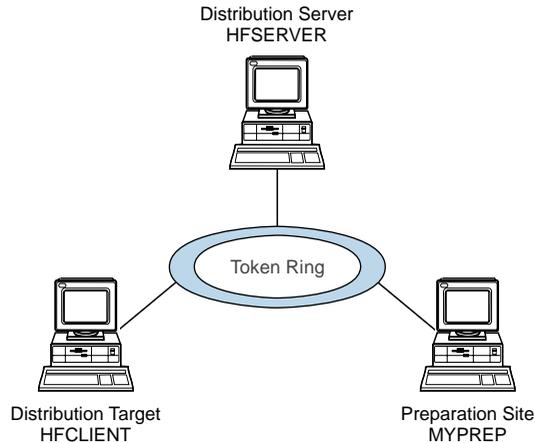


Figure 118. Software distribution environment

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 and Configuring the TME 10 Software Distribution Server for Windows NT or 2000" on page 53 for a detailed explanation of installing and configuring a TME 10 Software Distribution for Windows NT 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.

Distribution Database Recovery Tool

This tool is part of the distribution server. It provides a recovery procedure that you can run on the distribution server catalog when a specific error occurs.

Preparing an Application in Basic Mode

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, run **FNDRECV.CMD**.

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. It reclaims unused space resulting from deletion of entries from the database.

Before starting this tool, stop the distribution server. To run it, type FNDDBCMP.EXE at a command prompt, and press Enter.

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 your own backup repository.

Scenario 1: Preparing an Application in Basic Mode

In this scenario, you add a software object to the software library and copy it into the catalog at the distribution server.

Before beginning software preparation, perform these steps:

- 1 Create the directory sample on the **F** drive of the preparation site workstation
- 2 Create the files sample1.dat and sample2.dat under the directory sample

To add an application, such as the above-mentioned sample product, to the software library, perform the following steps starting from the preparation workstation:

- 1 Double-click on the **TME 10 Software Distribution** icon.

The TME 10 Software Distribution Common - Icon View folder appears, as shown in Figure 119 on page 203.

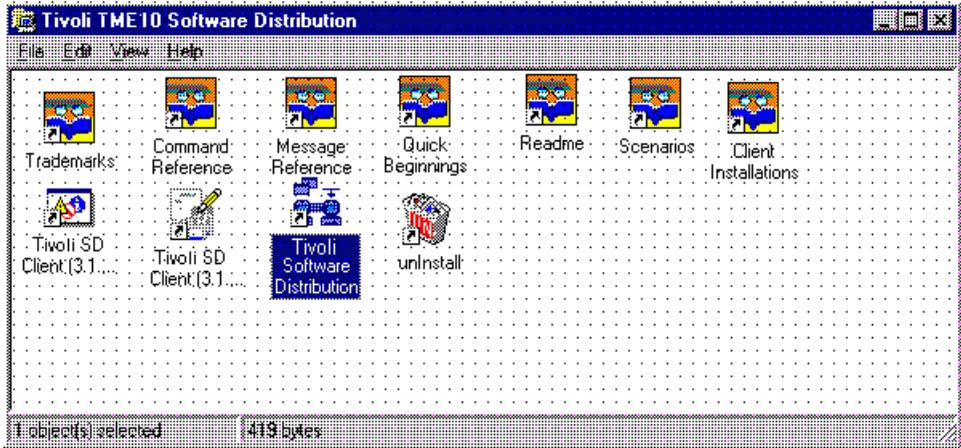


Figure 119. TME 10 Software Distribution common - icon view folder

- 2 Double-click on the **SW Preparation** icon.

The Software Object Profiles window appears, as shown in Figure 120.



Figure 120. Software Object profiles window

- 3 Select **Software** from the menu bar.
- 4 Select **Create new...** from the **New** choice in the pull-down menu.

The New Software Profile window appears, as shown in Figure 121 on page 204.

Preparing an Application in Basic Mode

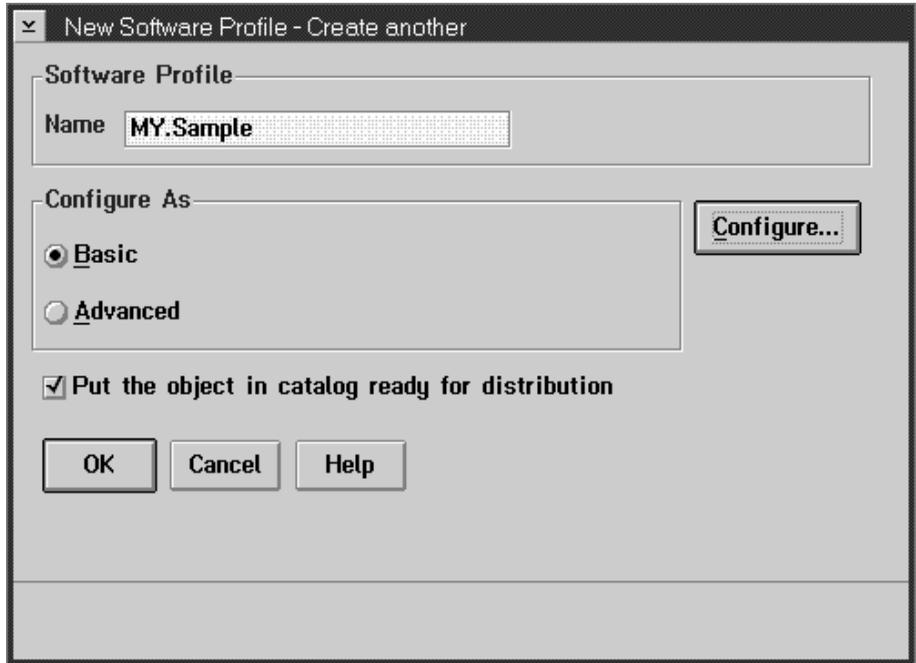


Figure 121. New software profile window

- 5 For **Name**, enter **MY.Sample**.
- 6 Select **Basic** in the **Configure As** group to a basic creation of a software profile.
- 7 Select **Configure...**

The MY.Sample - Basic Configuration notebook appears, as shown in Figure 122 on page 205.

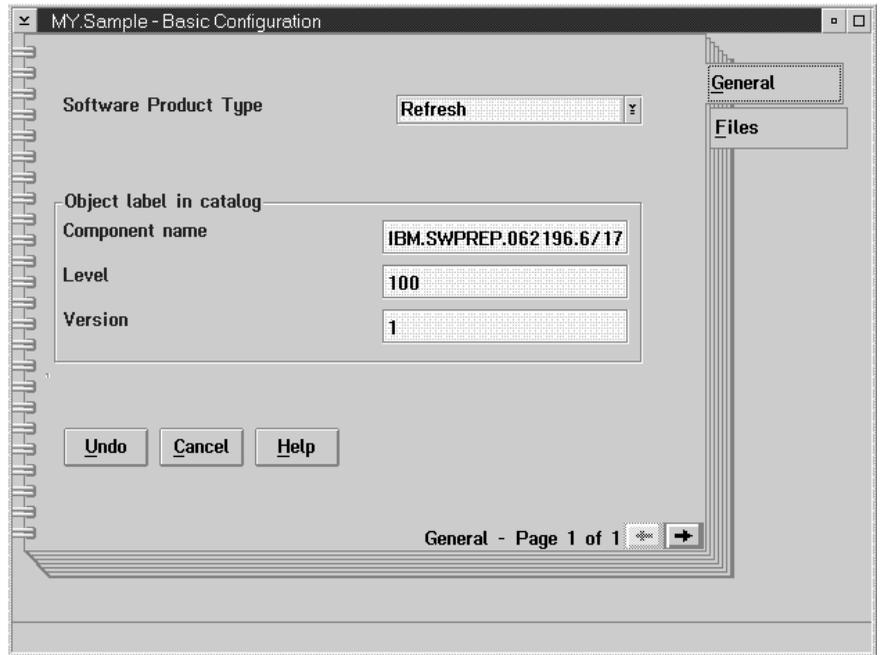


Figure 122. MY.Sample - basic configuration general Page

- 8 Accept the default values on the General page.
- 9 Select the **Files** tab.
The Files page appears.

Preparing an Application in Basic Mode

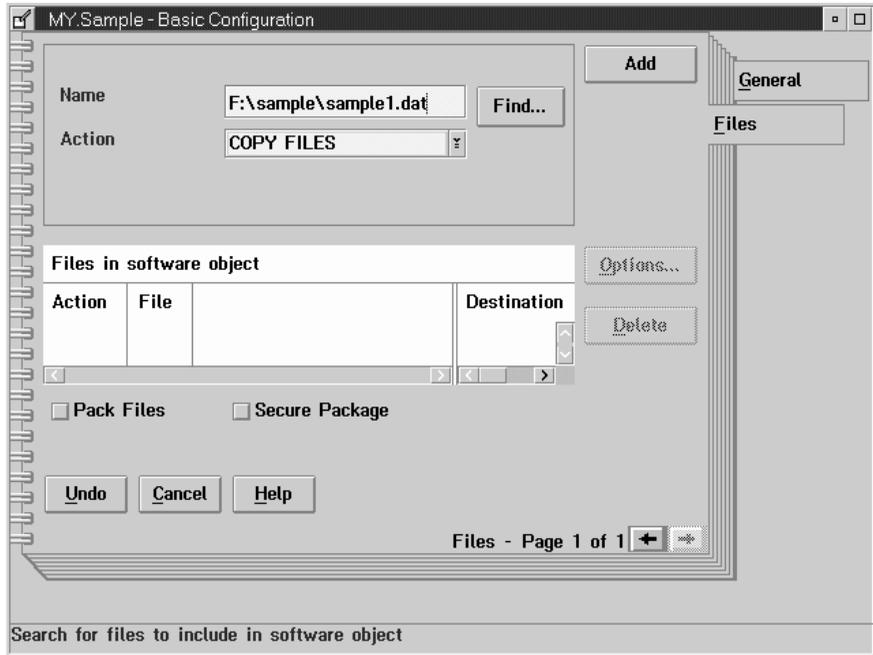


Figure 123. MY Sample - basic configuration files Page

- 10 On the Files page, choose the files to be included in the software object, and select **Find**.

The Open window appears, as shown in Figure 124.

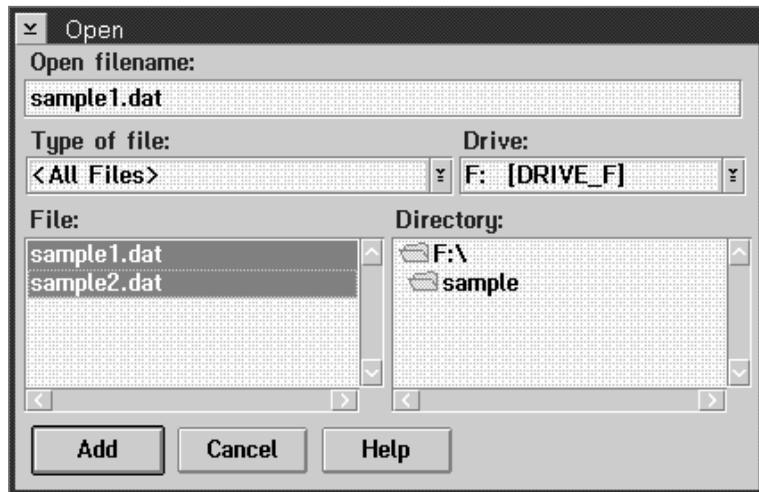


Figure 124. Open window

- 11 Choose the files that will be included in the software object, and select **Add**.
The selected files are included in the Files in software object container on the Files page, as shown in Figure 125.

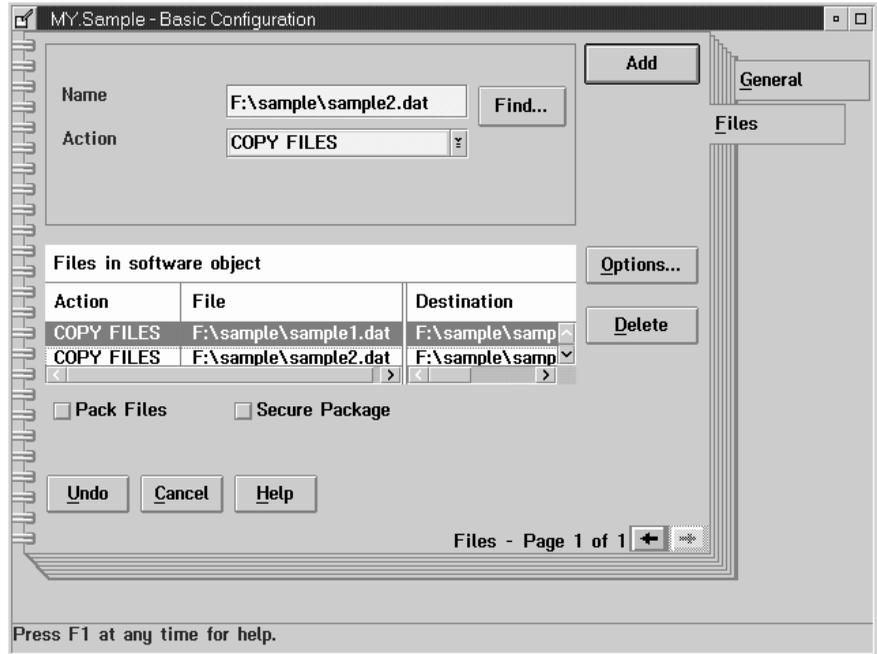


Figure 125. MY Sample - basic configuration files page

- 12 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 select **Options**.
The File Options window appears, as shown in Figure 126 on page 208.

Preparing an Application in Basic Mode

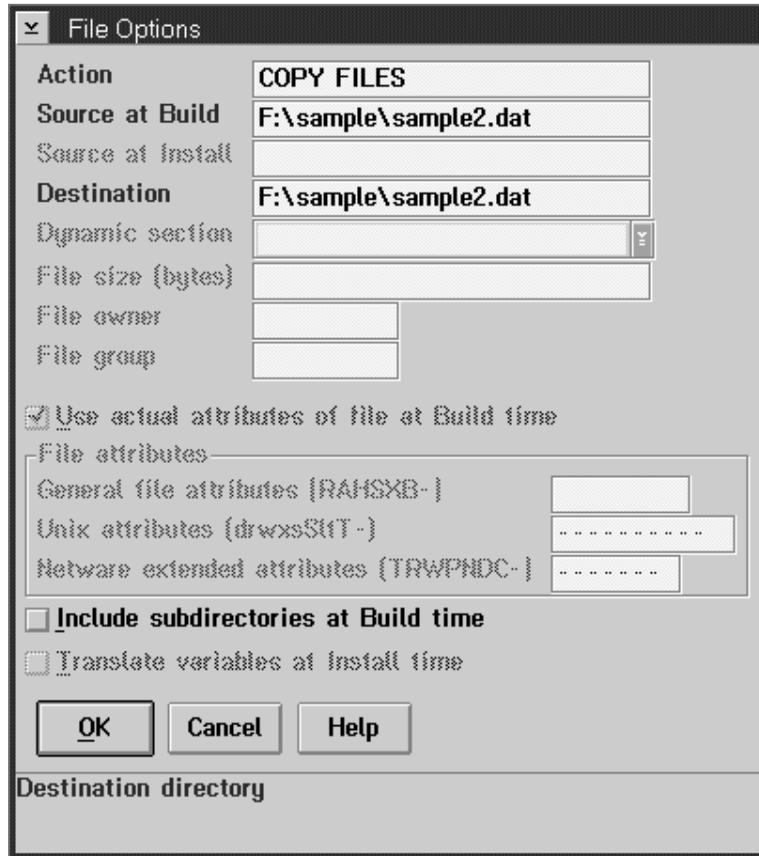


Figure 126. File options window

- 13 Change the destination of **sample1.dat** from **F** to **C**, by changing the installation path in the **Destination** entry field.
- 14 Select **OK**.
- 15 Repeat Steps 12, 13, and 14 for the **sample2.dat** file.
- 16 Close the notebook.

The New Software Profile window appears again, as shown in Figure 127 on page 209.

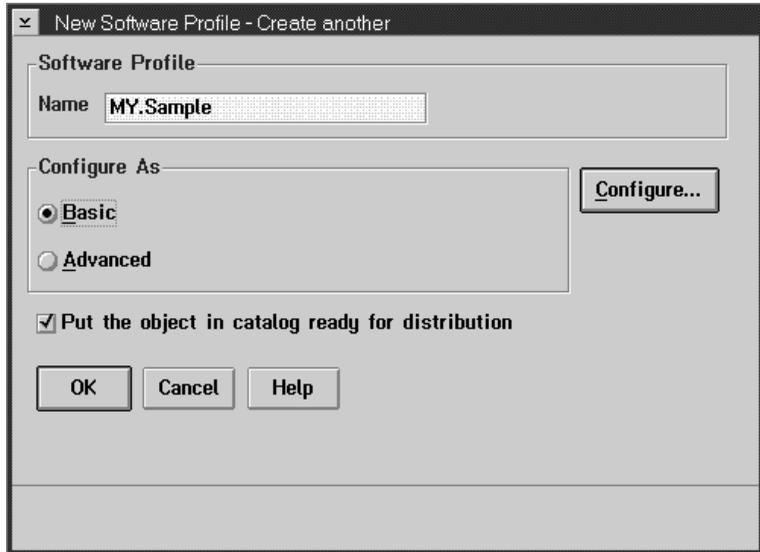


Figure 127. New software profile window

- 17 Leave the check box checked, and select **OK** to build the software object and put it into the catalog.

A progress indicator appears, as shown in Figure 128.

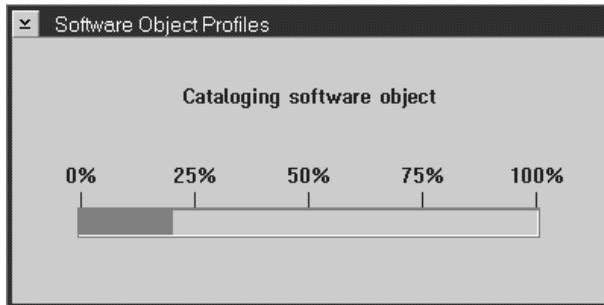


Figure 128. Software Object profiles progress indicator

A Software Object Profiles informational message appears, as shown in Figure 129 on page 210.

Preparing an Application in Basic Mode



Figure 129. Software Object profiles informational message

18 Select **OK**.

The Software Object Profiles window containing the **MY.Sample** software object profile that you created appears, as shown in Figure 130.

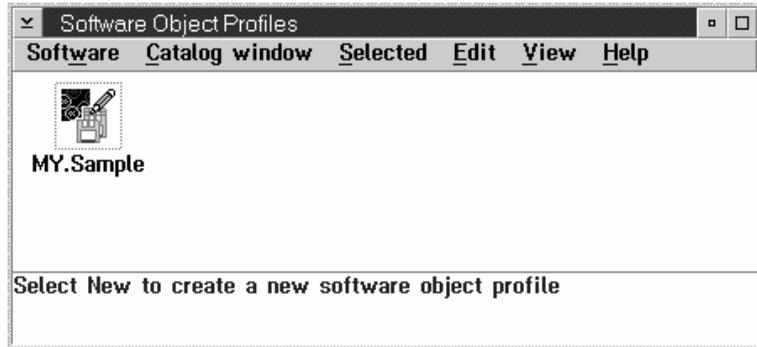


Figure 130. Software Objects profile Window

Now the **MY.Sample** software object is ready to be installed. See the online *Command Line Reference* for information on how to install the software object.

Scenario 2: Preparing an Application in Advanced Mode

During this scenario you will perform these tasks:

- Use the Remote Directory function to add a software object to the TME 10 Software Distribution software library and copy it into the catalog at the distribution server.
- Add a dynamic software object to the TME 10 Software Distribution software library and copy it into the catalog at the distribution server.

Adding a Software Object Profile to the Software Library With Remote Directory

For the purposes of this scenario, you first 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 add an application, such as the above sample product, to the software library, perform the following steps starting from the TME 10 Software Distribution Service Manager window:

- 1 Double-click on the **Tivoli Software Distribution** icon.

The Software Distribution Window appears.

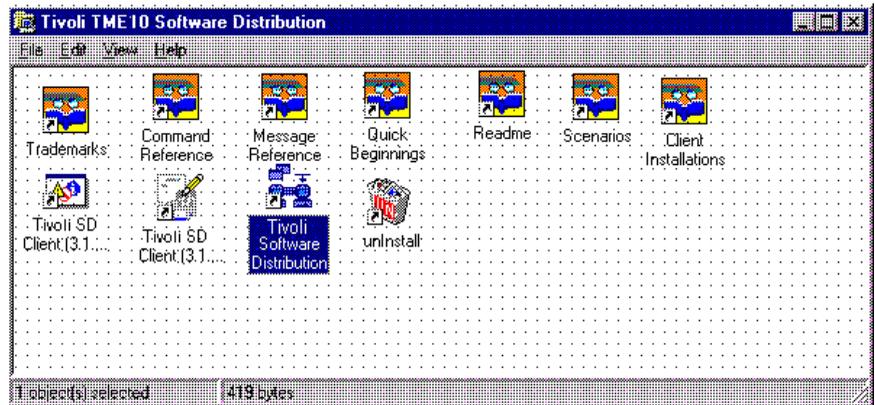


Figure 131. TME 10 Software Distribution common - icon view folder

- 2 Double-click on the **Software Preparation** icon.

The Software Object Profiles window appears, as shown in Figure 132 on page 212.

Preparing an Application in Advanced Mode



Figure 132. Software Object profiles window

- 3 Select **Software** from the menu bar.
- 4 Select **Create new...** from the **New** choice in the pull-down menu.
The New Software Profile window appears, as shown in Figure 133.

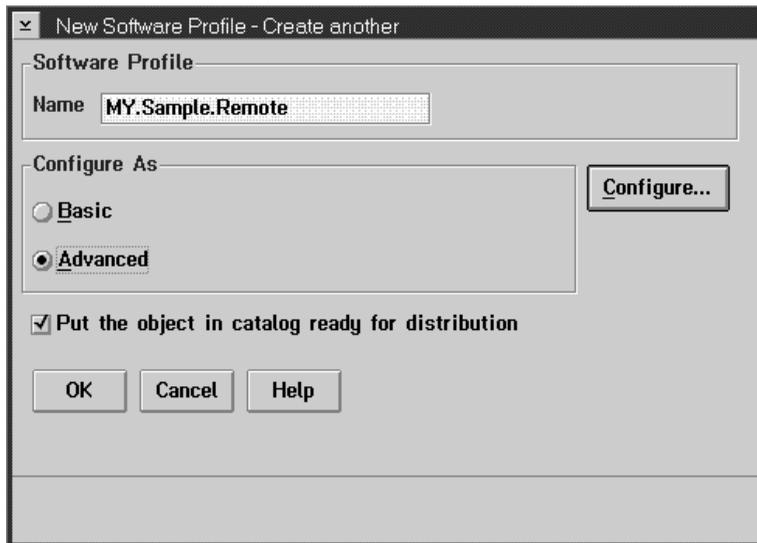


Figure 133. New software profile window

- 5 For **Name**, enter **MY.Sample.Remote**.
- 6 Select **Advanced** in the **Configure As** group to do an advanced creation of a software profile.
- 7 Select **Configure...**

The My.Sample.Remote - Advanced Configuration notebook appears, as shown in Figure 134 on page 213.

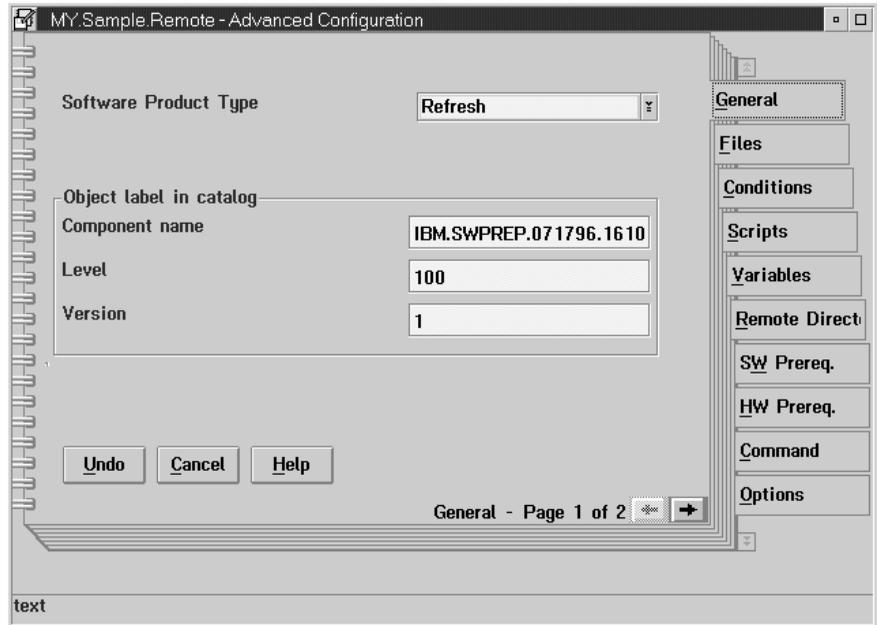


Figure 134. *My.Sample.Remote* - advanced configuration general page

8 Leave the defaults in the General page.

9 Select the **Remote Directory** tab.

The Remote Directory page appears, as shown in Figure 135 on page 214.

Preparing an Application in Advanced Mode

Server name	Exported directory	Mount drive	Mount paramete
HFSERVER	Sample	F:	

Figure 135. *My.Sample.Remote - advanced configuration remote directory page*

On the Remote Directory page, choose the remote directory that contains the files to be included in the software object. These files will be downloaded using a mount process at installation time.

- 10 Fill in the entry fields that are contained in the Information to mount a remote directory group, as shown in Figure 135.

Server name The name of the distribution server where the files are stored.

Exported directory The remote directory that will be mounted. It contains the files for the software object.

You may want an agent workstation to mount a directory of a Windows NT server remotely when the agent has been started and no user is logged on to the agent workstation. To make this possible, perform the following operations on the Windows NT server:

- a. Start the registry editor by entering the command REGEDT32.EXE.



Be careful when you use REGEDT32 not to change any keys that are not specified

here. Changing keys by mistake could request a Windows NT reinstallation. It is good practice, in any case, to make a backup of the file REGEDT32.CNT before making any changes with the registry editor.

- b. From HKEY_LOCAL_MACHINE select:

```

\SYSTEM
  \CurrentControlSet
    \Services
      \LanmanServer
        \Parameters
    
```

- c. From the Edit menu, select **Add Value**.

- d. In the Value Name field, write **RestrictNullSessAccess**.

When the installation is complete, delete RestrictNullSessAccess to avoid a security exposure. RestrictNullSessAccess allows all machines that are connected to the Windows NT server to share files.

- e. In the Data Type field, select **REG_DWORD** and then select **OK**.
- f. In the DWORD editor field enter **0**, and then select **OK**.
- g. Exit REGEDT32, and then shut down and restart Windows NT

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.

11 Select **Add**.

The mount information is included in the Exported directories list container.

12 Select the **Files** tab. The Files page appears. In the **Name** field, enter the name of one of the remote objects (sample1.dat).

Select **Add**.

Repeat this step for the sample2.dat object.

13 Select COPY REMOTE FILES from the **Action** list.

Preparing an Application in Advanced Mode

- 14 Close the notebook. The New Software Profile window appears again, as shown in Figure 136.

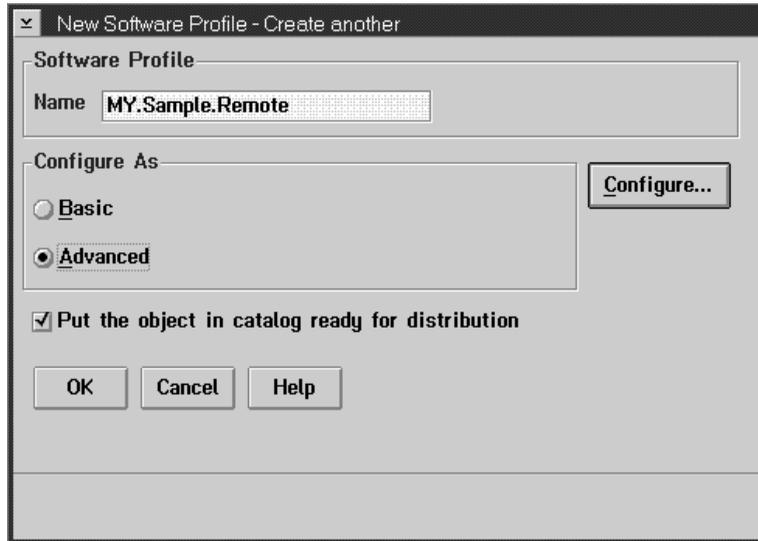


Figure 136. New software profile window

- 15 Leave the check box checked, and select **OK**. to build the software object and put it into the catalog.

A progress indicator appears, as shown in Figure 137.

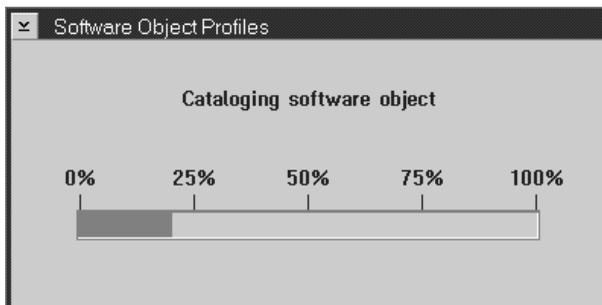


Figure 137. Software Object profiles progress indicator

A Software Object Profiles informational message appears as shown in Figure 138 on page 217.



Figure 138. Software Object profiles informational message

16 Select **OK**.

The Software Object Profiles window that contains the **My.Sample.Remote** object you created appears, as shown in Figure 139.

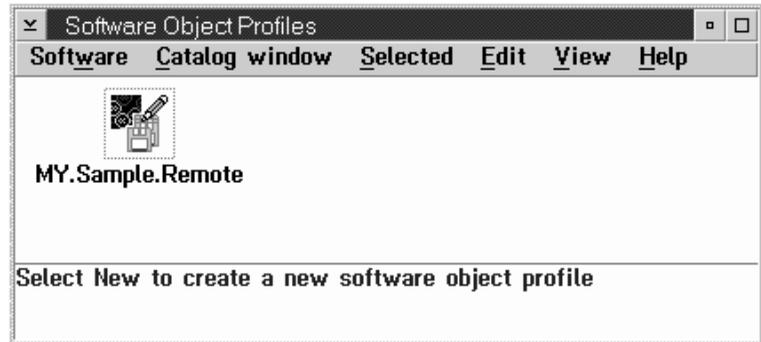


Figure 139. Software Objects profile window

Now this software object is ready to be installed. See the online *Command Line Reference* for information on how to install the software object.

Adding a Dynamic Software Object to the Software Library

A dynamic software object enables the user 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 enables you to install two files (**sample1.dat** and **sample2.dat**) on two targets (one with a disk of 150 MB and another with a disk of 200 MB) with the same **Install** operation.

Use the sample product you created at the beginning of Scenario 1.

- 1 Create the **MY.Sample.Dynamic** software profile in advanced mode, as shown in the previous section.

Preparing an Application in Advanced Mode

- 2 Open the My.Sample.Dynamic - Advanced Configuration notebook. It appears in Figure 140

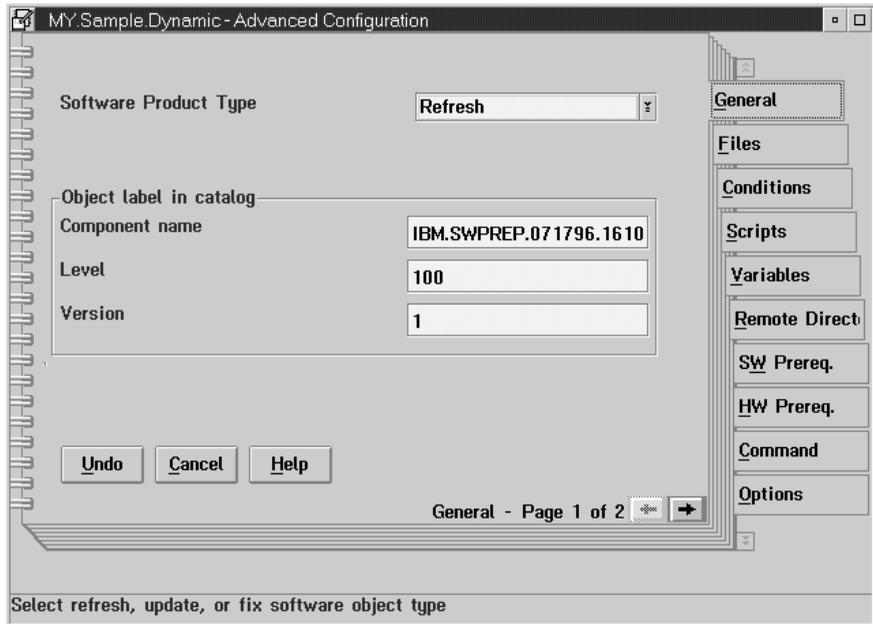


Figure 140. My.Sample.Dynamic - advanced configuration general page

- 3 Accept the defaults on the General page.
- 4 Select the **Files** tab.
The Files page appears, as shown in Figure 141 on page 219.



Figure 141. My.Sample.Dynamic - advanced configuration files page

Now you create two dynamic sections:

- The first section to install the **sample1.dat** file on a workstation with a disk space of 150 MB.
- The second section 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 Select **Open....**

The Dynamic Section - Create notebook appears, as shown in Figure 142 on page 220.

Preparing an Application in Advanced Mode

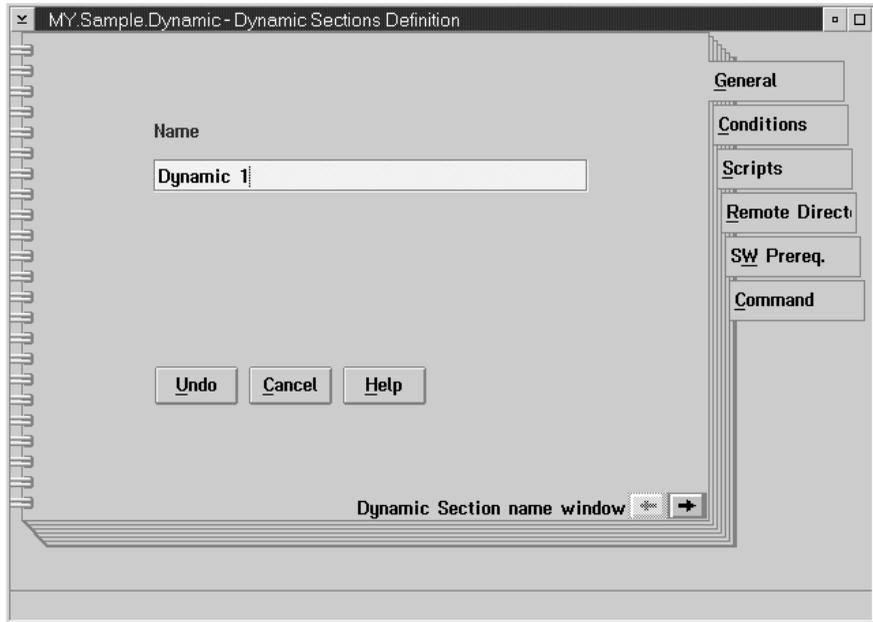


Figure 142. Dynamic section - create notebook - general page

8 For **Name**, enter **Dynamic 1**.

9 Click on the **Conditions** tab.

The Condition page appears, as shown in Figure 143 on page 221.

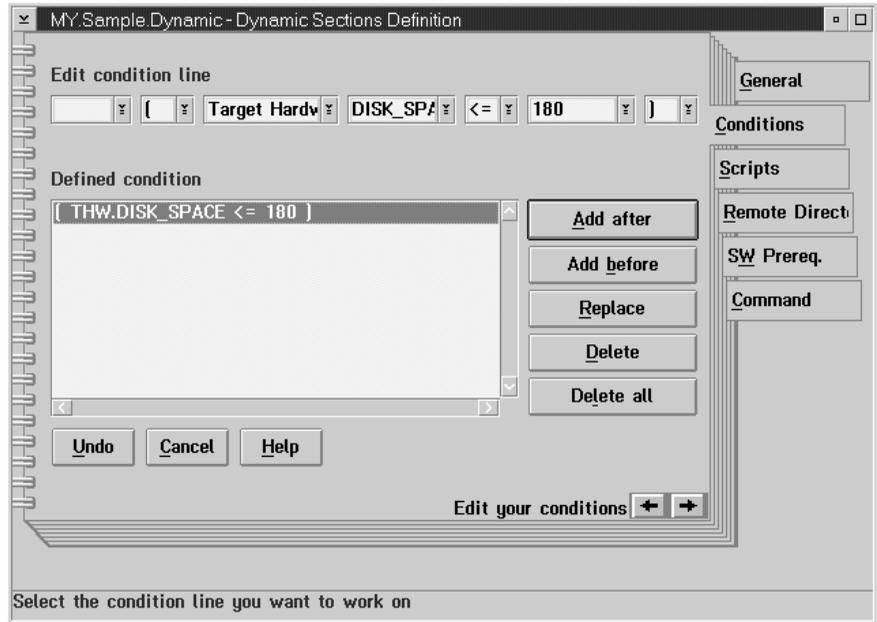


Figure 143. Dynamic section - create notebook - condition page

- 10 Fill in the entry fields contained in the Edit condition line group, to define the rule of the first condition [DISK_SPACE <= 180], as shown in Figure 143.
- 11 Select **Add after**.
The rule appears in the Defined condition list.
- 12 Close the notebook.
The My.Sample.Dynamic - Advanced Configuration notebook reappears.
- 13 Select **Add**.
The **sample1.dat** file and the corresponding **Dynamic 1** dynamic section are displayed in the Files software object container.
- 14 Enter **sample2.dat** in the Name field.
- 15 Accept the default in the Action list.
- 16 Select **Open...**
The Dynamic Section - Create notebook appears, as shown in Figure 144 on page 222.

Preparing an Application in Advanced Mode

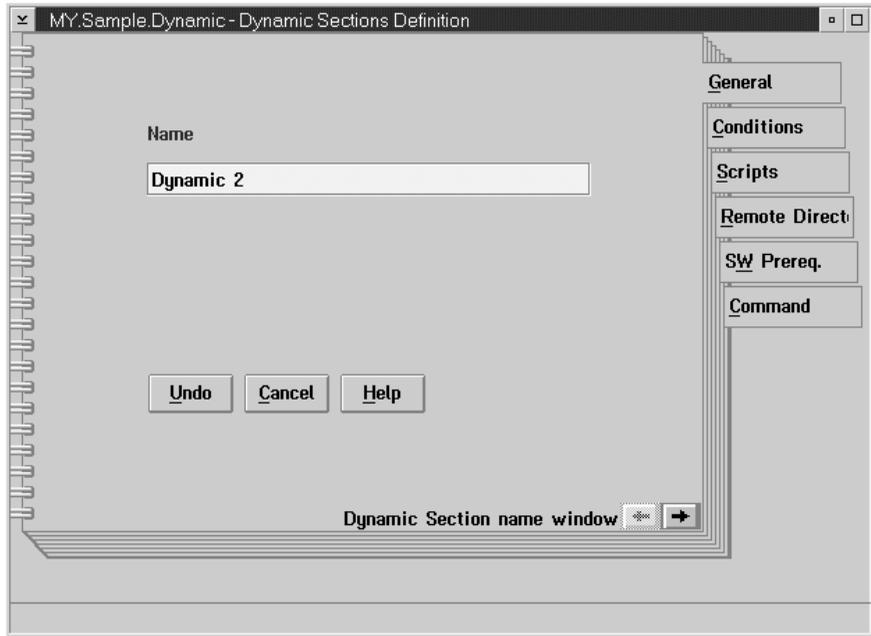


Figure 144. Dynamic section - create notebook - general page

17 For **Name**, enter **Dynamic 2**.

18 Click on the **Conditions** tab.

The Condition page appears, as shown in Figure 145 on page 223.

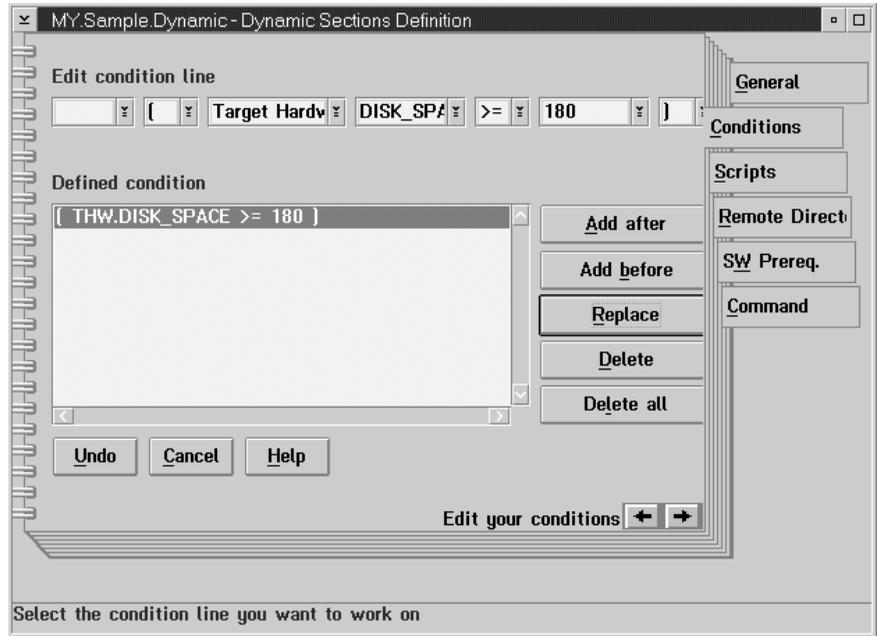


Figure 145. Dynamic section - create notebook - condition page

- 19 Fill in the entry fields contained in the Edit condition line group, to define the rule for the second condition [DISK_SPACE >= 180], as shown in Figure 145.
- 20 Select **Add after**.
The rule appears in the Defined condition list.
- 21 Close the notebook. The My.Sample.Dynamic - Advanced Configuration notebook appears again.
- 22 Select **Add**.
The **sample2.dat** file and the corresponding **Dynamic 2** dynamic section are displayed in the Files software object container as shown in Figure 146 on page 224.

Preparing an Application in Advanced Mode



Figure 146. MY.Sample.Dynamic - advanced configuration files page

23 Close the notebook.

The New Software Profile window appears again, as shown in Figure 147 on page 225.

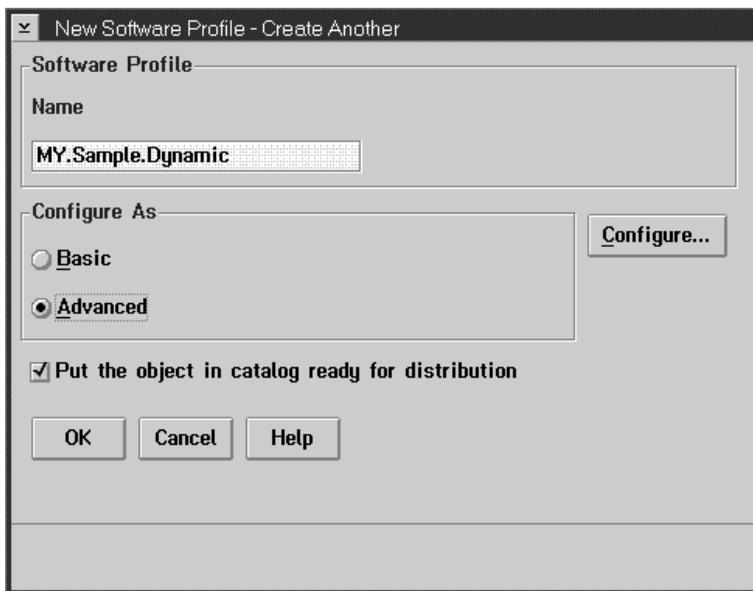


Figure 147. New software profile window

- 24** Leave the check box checked, and select **OK** to build the software object and put it into the Catalog.

A progress indicator appears, as shown in Figure 148.

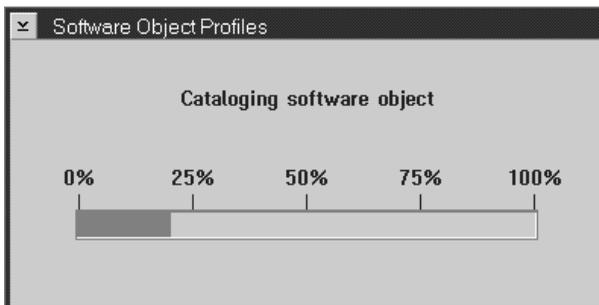


Figure 148. Software Object profiles progress indicator

A Software Object Profiles informational message appears as shown in Figure 149 on page 226.

Preparing an Application in Advanced Mode



Figure 149. Software Object profiles informational message

25 Select **OK**.

The Software Object Profiles window that contains **My.Sample.Dynamic** object you created.

Now this dynamic software object is ready to be installed. See the online *Command Line Reference* for information on how to install the dynamic software object.

Scenario 3: Replicating an Installation with DiskCamera

XLATOR is an application to make available to other users. XLATOR does not have an installation program. To install it, copy the files to a directory and some subdirectories, and modify 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 files¹ and workplace objects² *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.

Taking a “Before” Picture

You first take a “before” picture of the E drive, install XLATOR on it, 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 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:

-
- 1 On OS/2, the default list of files monitored by DiskCamera is OS2INIT.COM, OS2SYS.INI, OS2.INI, STARTUP.COM, and CONFIG.SYS.
On Windows 3.11 clients, CONFIG.SYS, AUTOEXEC.BAT, and all .INI files are monitored. On Windows 2000, Windows NT, Windows 98, Windows 95, and Windows 3.11 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.
 - 2 On OS/2, each workplace object (icon, folder, launch pad, etc.) on the current desktop is monitored, only if it was created with an object ID. DiskCamera does not monitor workplace objects created without an object ID.
On Windows 98 and Windows 95, 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, perform the following 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 Take the “after” picture.

Replicating an Installation with DiskCamera

- 1 Double-click on the **Software Distribution** icon.

The TME 10 Software Distribution - Icon View folder appears, as shown in Figure 150.

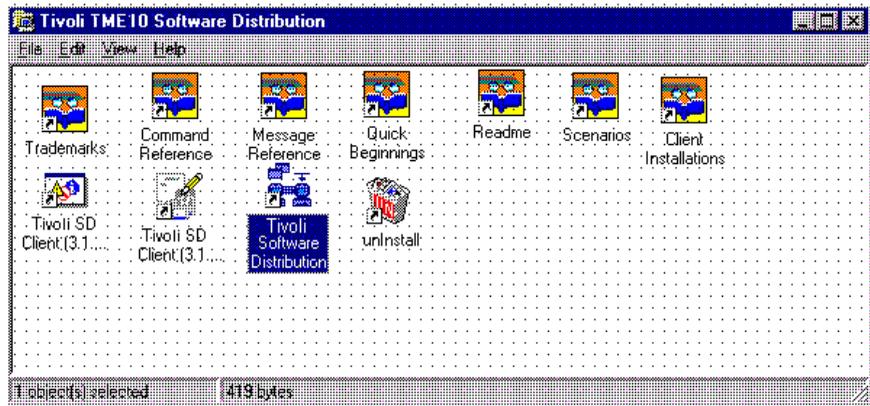


Figure 150. TME 10 Software Distribution - icon view folder

- 2 In the TME 10 Software Distribution window, double-click on the **Software Preparation** icon.

The Software Object Profiles window appears, as shown in Figure 151.



Figure 151. Software Object profiles window

- 3 Select **Software** from the menu bar.
- 4 Select **Using DiskCamera...** from the **New** choice in the pull-down menu.

The Create a software object profile using DiskCamera window appears, as shown in Figure 152 on page 229.



Figure 152. Create a Software Object profile using DiskCamera window

- 5 Select the first button (it is the only one available).

The Take a picture of a disk drive window appears, as shown in Figure 153.



Figure 153. Take a picture of a disk drive window

- 6 Select the **E** drive and select **OK**. The first picture starts. This operation could take several minutes.
- 7 An informational message informs you when the disk picture has been successfully completed. Select **OK**.

The Create a software object profile using DiskCamera window appears 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.

Replicating an Installation with DiskCamera

Now you must install the new application at the preparation site.

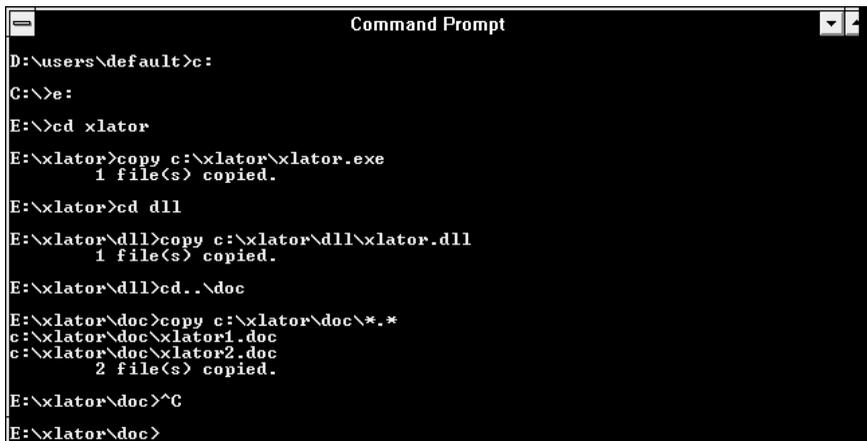
Installing the Application at the Preparation Site

XLATOR is stored on the C drive, under the directory 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, perform the following steps:

- 1 Create directory XLATOR, with subdirectories DLL and DOC, on the E drive.
- 2 From a MS-DOS prompt, copy the files to the E drive, as shown in Figure 154.



```
Command Prompt
D:\users\default>c:
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>^C
E:\xlator\doc>
```

Figure 154. Copying files to the DiskCamera drive

- 3 Using the system editor, edit the system environment variables, adding XLATOR to the Value field, as shown in Figure 155 on page 231, and save the changes.

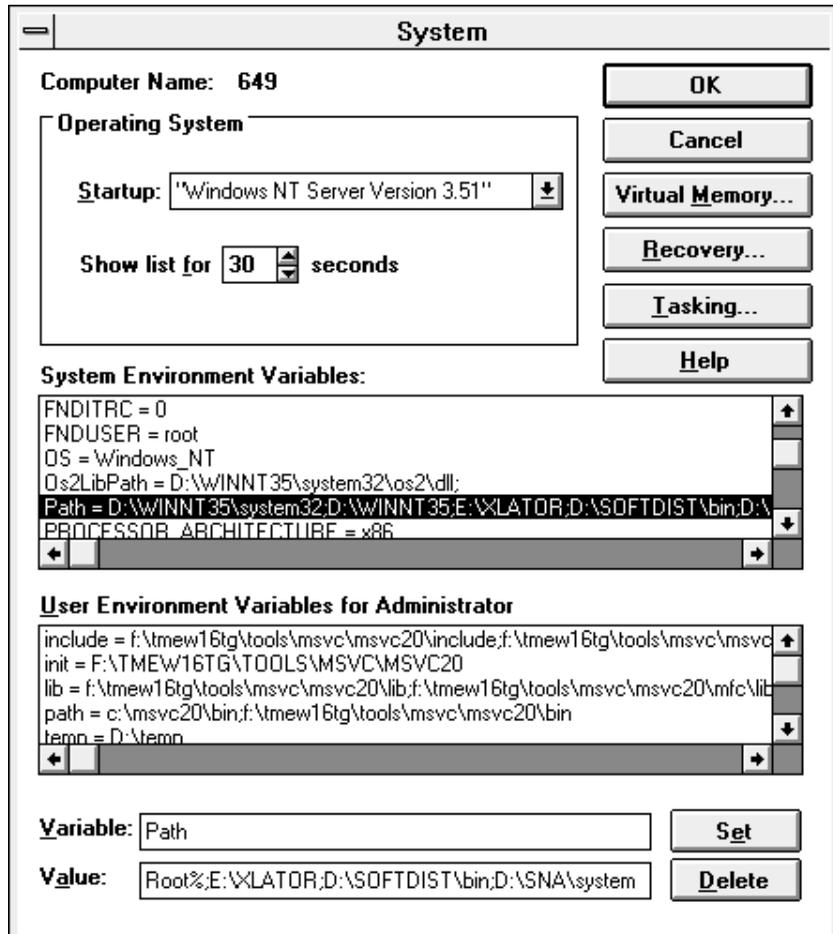


Figure 155. Editing the system environment variables

Now you are ready to take the “after” picture.

Taking an “After” Picture

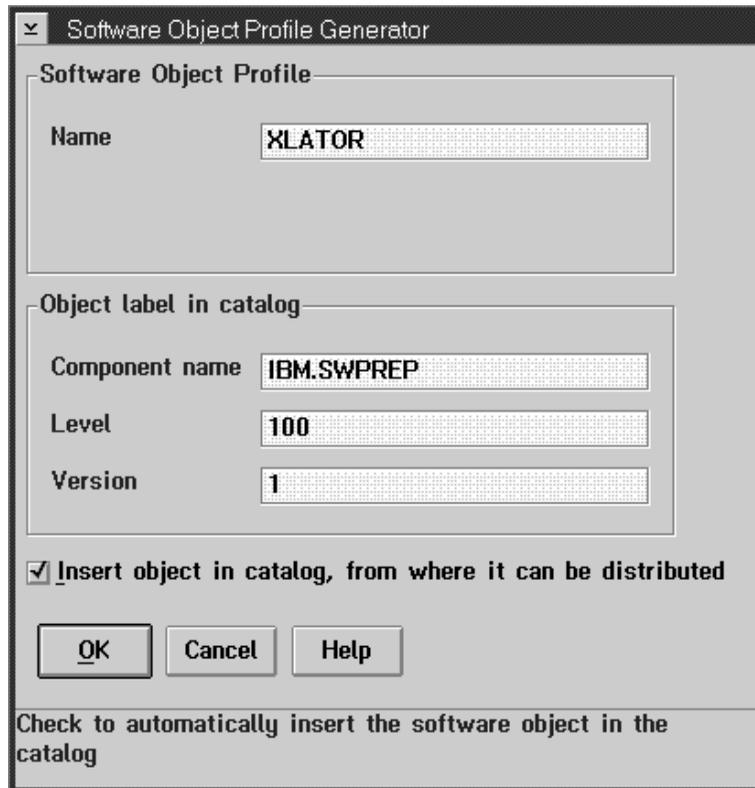
After the installation is completed, the next step is to take the “after” picture of the installation drive and the files being monitored. Note that you must be sure that the installation is complete before you take the “after” picture. If you need to change the installation, you will have to restart the DiskCamera process from the beginning.

It is also important to be sure all the 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, perform the following steps:

Replicating an Installation with DiskCamera

- 1 In the Create a software object profile using DiskCamera window (shown in Figure 152 on page 229), click on the second button to notify DiskCamera that the installation has completed.
A confirmation message appears.
- 2 Select **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 completed.
- 3 In the Create a software object profile using DiskCamera window (shown in Figure 152 on page 229), select the third button to generate the software object profile.
The Software Object Profile Generator window appears.
- 4 Fill in the fields of this window, as shown in Figure 156.



The screenshot shows a dialog box titled "Software Object Profile Generator". It contains two main sections for input:

- Software Object Profile:** A text field labeled "Name" containing the text "XLATOR".
- Object label in catalog:** Three text fields: "Component name" containing "IBM.SWPREP", "Level" containing "100", and "Version" containing "1".

Below these sections is a checked checkbox labeled "Insert object in catalog, from where it can be distributed". At the bottom are three buttons: "OK", "Cancel", and "Help". A footer note reads "Check to automatically insert the software object in the catalog".

Figure 156. 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.

- 5 To catalog the software object, select the Insert object in catalog check box (the default is "checked").

In this example, if you want only to create the software object profile and catalog it later, remove the check mark from the check box.

- 6 Select **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. This operation could take several minutes.

A progress indicator appears, as shown in Figure 137 on page 216.

An informational message appears, as shown in Figure 157.

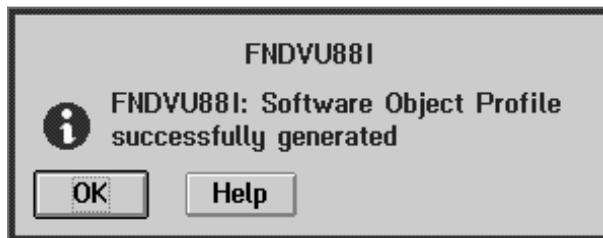


Figure 157. Informational message

- 7 Select **OK**.

The Create a software object profile using DiskCamera window appears, as shown in Figure 158 on page 234.

Replicating an Installation with DiskCamera

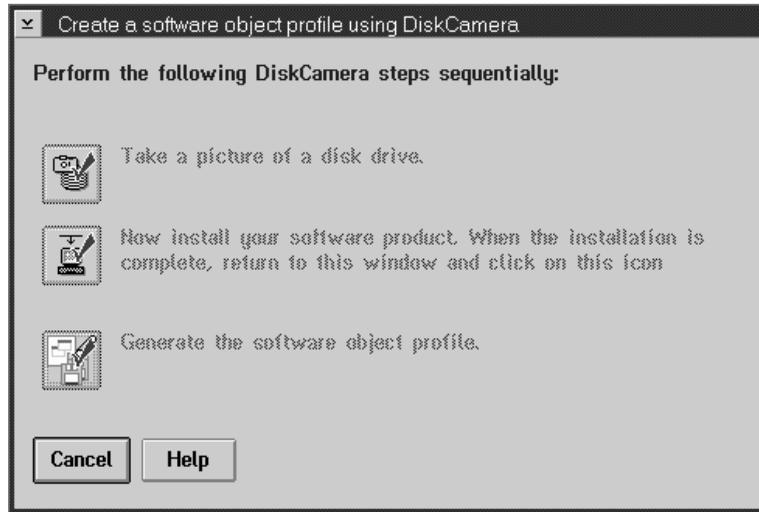


Figure 158. Create a Software Object profile using DiskCamera window

Now all the buttons are checked. This means that the DiskCamera process has completed.

8 Select **Cancel**.

The creation of the XLATOR software object profile has completed.

The icon for XLATOR appears in the Software Object Profiles window, as shown in Figure 159.

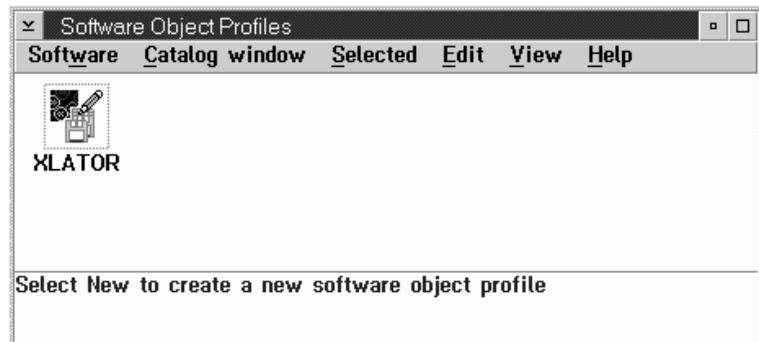


Figure 159. Software Object profiles window with XLATOR added

Now you will check the files associated with the XLATOR software object profile. You will also change the drive on which the XLATOR software object will be installed.

9 Double-click on the XLATOR icon in the Software Object Profiles window.

The XLATOR - Advanced Configuration notebook appears, as shown in Figure 160 on page 235.

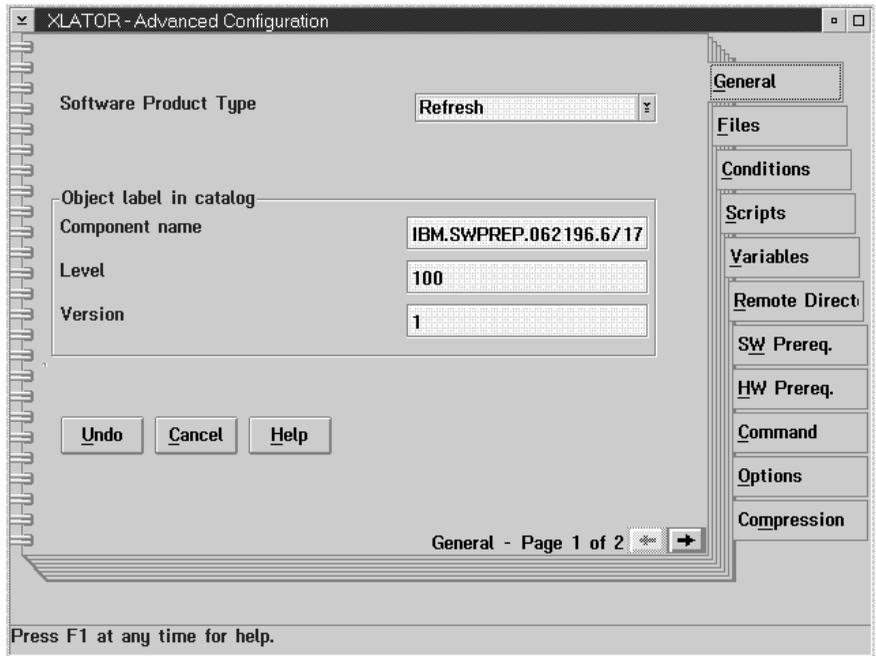


Figure 160. XLATOR - advanced configuration notebook

- 10** Click on the **Files** tab to see the list of files that have been included in the software object. The list appears in the Files in software object container, as shown in Figure 161 on page 236. You can use this page to add, update, or delete files to the software object.

Replicating an Installation with DiskCamera



Figure 161. XLATOR - advanced configuration notebook showing the files page

- 11 To check the variables associated with the software object, click on the **Variables** tab.

The **Variables** page appears, as shown in Figure 162 on page 237.

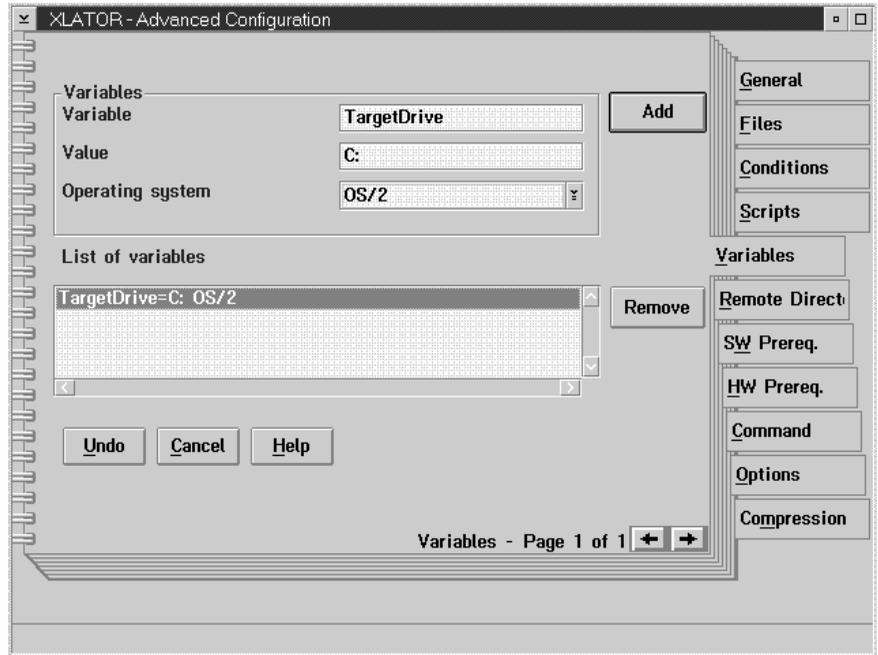


Figure 162. XLATOR - advanced configuration notebook showing the variables page

In this case, the only variable is the target drive on which XLATOR is to be installed.

To change the target drive, perform the following steps:

- a Select **TargetDrive=E** in the List of variables.
- b Change **E** to **C** in the **Value** entry field.
- c Select **Windows NT** in the Operating system list.
- d Select **Add**.

The **TargetDrive=C** item appears in the List of variables.

- 12 Close the XLATOR - Advanced Configuration Notebook.

Cataloging the Software Object

After you have confirmed that XLATOR contains the correct files and you have changed 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 163.

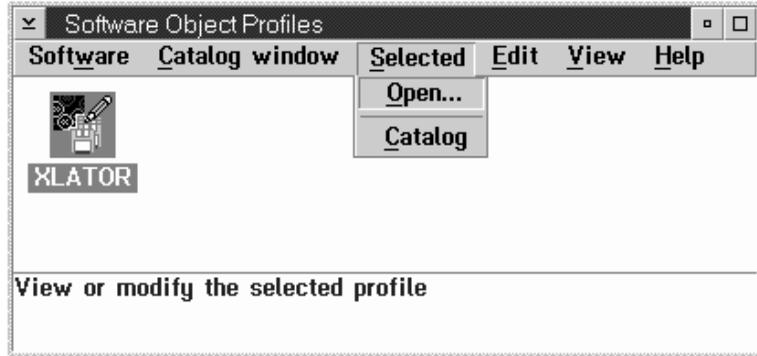


Figure 163. Software Object profile window

A progress indicator appears, as shown in Figure 164.

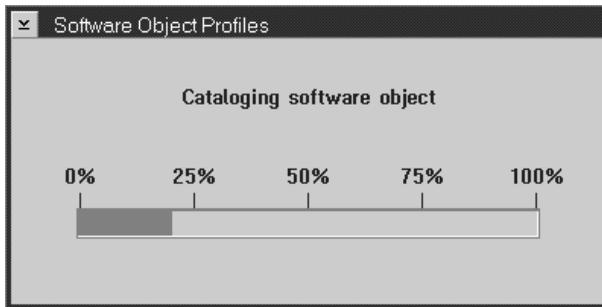


Figure 164. Software Object profiles progress indicator

An information message appears, as shown in Figure 165 on page 239.



Figure 165. Software Object profiles information message

3 Select **OK**.

An object for XLATOR is created in the TME 10 Software Distribution Catalog.

Note that the preparation site (MYPREP) and the target workstation (HFCLIENT) should have the same fix level of the operating system installed.

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, "Implementing Inventory Discovery" on page 243
- Appendix B, "Replacing the Quiesce Check" on page 249
- Appendix C, "Writing Change Control Scripts" on page 251
- Appendix D, "Writing User Exits" on page 255
- Appendix E, "Setting Environment Variables" on page 269

Appendix A. 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 dinv`, 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:

<code>fn dtkinv</code>	Stores discovered installation parameters
<code>fn dswinv</code>	Stores discovered software information
<code>fn dhwinv</code>	Stores discovered hardware information

“Mapping to the TME 10 Software Distribution Inventory Files” on page 245 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 245. 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 166 on page 246 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:   WINDOWS NT
  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 166. 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 Windows NT 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 246.

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 248). 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 75.

Appendix B. 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 167 on page 250 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 167. Example of a check script used for quiesce checks

Appendix C. 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 D. 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`, `fndssex.c`, and `fndsx.c`
- Make file, used when compiling and linking the user exit code: `ntadx.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`.

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`.

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*.

Writing User Exits

This user exit is located in the source file `fndcxmo.c`.

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`.

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.

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`.

Writing User Exits

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 `fndswcms` 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 `fndswcms` 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

- 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

```
*****/
```

Writing User Exits

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

Compilation Method

After compilation, the user exits are contained in three DLLs: FNDCX.DLL, FNDSX.DLL, and FNDSS.DLL. Three make files, FNDCX.MAK, FNDSX.MAK, and FNDSS.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>
```

_____ End of General-Use Programming Interface _____

Appendix E. Setting Environment Variables

To accomplish the tasks listed below, you must set the required environment variables.

To set an environment variable permanently:

If you are using TME 10 Software Distribution for Windows 95

Save the environment variable in the AUTOEXEC.BAT file and reboot the workstation. For example, to set ACT_ON_TARGET to 1 permanently, set the following in the AUTOEXEC.BAT file:

```
set ACT_ON_TARGET=1
```

If you are using TME 10 Software Distribution for Windows NT

Save the environment variable in the System Properties window. For example, to set ACT_ON_TARGET to 1 permanently, perform the following steps:

1. Click **Start** and then select **Settings ► Control Panel ► System**.
2. In the System Properties window, select the **Environment** tab.
3. In the **Variable** field, type ACT_ON_TARGET and in the **Value** field, type 1.
4. Click **Set** to add the environment variable to the list.
5. Click **OK** to close the window.

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.

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

Setting Environment Variables

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.

To save memory usage, if you do not have a constant traffic rate, do not set this variable.

Shutting Down the Product Using the FNDBOOT.cmd Command

Set the FNDBOOT environment variable to shut down the product using the FNDBOOT.cmd command. If you do not set this variable, the product is shut down using the InitiateSystemShutdown process.

Enabling Thread Termination During the Closure of the NVDMGI.exe Process

Set the FNDNTGUI environment variable to YES to enable thread termination when the NVDMGI.exe process is closing.

Hiding Black Windows Executing Script Files

Set the FNDHIDE environment variable to YES to hide black windows when you execute script files. You can use this variable in the TME 10 Software Distribution for Windows NT, TME 10 Software Distribution for Windows 95, and TME 10 Software Distribution for Windows 98 environments.

Disabling the Shutdown Pop-up Menu During an Unattended Uninstallation

Set the FNDNOOK environment variable to YES to not display the shutdown pop-up menu that will require user interaction, if you are uninstalling the product in unattended mode.

Index

A

access keys 41
 ACT_ON_TARGET 269
 active client 5
 active, software object status 36
 adding target parameters, user exit 260
 adding user parameters, user exit 262
 administrator authorization profile 39
 administrator, change control 9
 AIX client
 allocation failure retry time, SNA/DS configuration file 106
 API trace file size, base configuration parameter 76
 ASCII files, translation 34
 attended installation
 TME 10 Software Distribution server 53
 authorization profiles, user 39
 authorizations, user 89
 authorize, base configuration parameter 76
 AUTOEXEC.BAT
 monitoring in a DiskCamera installation 227
 automatic registration
 client 95
 definition 20
 keywords 95
 target 95
 automatic target inventory, base configuration parameter 76
 automatic target registration, base configuration parameter 77
 available, software object status 36

B

backlevel, software object status 36
 backup area, base configuration parameter 77
 base configuration file
 API trace file size 76
 authorize 76
 automatic target inventory 76
 automatic target registration 77
 backup area 77
 configuration 77
 connection window duration 78
 DACA idle time 78

base configuration file (*continued*)
 DACA retry time 78
 editing 75
 example 84
 file system 78
 inventory program 78
 LAN authorization 79
 log file size 79
 machine type 79
 max attempts 79
 max connections 79
 max local targets 79
 max requests 80
 max server conns 80
 max server targets 80
 max SNA/DS routes 80
 max STS 80
 max STS routes 80
 max targets 81
 max user interfaces 81
 message log level 81
 protocol 81
 repository 83
 server 83
 service area 83
 STS idle time 83
 STS retry time 83
 target password authentication 84
 TCP/IP port 84
 trace file size 84
 work area 84
 workstation name 84

books

INF files 44
 PDF files 46
 PostScript 45
 TME 10 Software Distribution products xxi
 TME 10 Software Distribution, Version 3.1.5 for Windows NT 44
 builder authorization profile 40
 builder, type of user 9

C

catalog
 cleanup tool 202

Index

- catalog (*continued*)
 - purpose 27
 - recovery tool 201
- catalog window 10
- CD-ROM
 - requirement for 50
 - SD4DOCPS directory 45
- change control
 - active status 36
 - administrator 9
 - available status 36
 - backlevel status 36
 - discovered status 36
 - 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 266
- CID (configuration, installation, and distribution)
 - installation method 27
- cleanup tool, database 202
- client, target type 92
- clients
 - automatic registration 95
 - fully disconnected 19
 - mobile 18
 - platforms supported 21
 - role in network 4
 - targets 17
 - user interface only 9
- command line interface
 - help 45
 - INF file 44
 - starting 13
 - stopping 13
- communication protocols
 - customizing a Windows NT or Windows 2000 server for 61
- communication protocols (*continued*)
 - supported 22
- communications programs
 - Communications Server,
 - MS SNA Server
 - Personal Communications
 - Communications Server
 - configuring 145
- components, TME 10 Software Distribution
 - adding additional 68
 - TME 10 Software Distribution server 55
 - updating 68
- compression, data 33
- compression, in software object 25
- CONFIG.SYS
 - monitoring in a DiskCamera installation 227
- configuration
 - for NetBIOS and IPX 61
 - keywords 75
- configuration file, SNA/DS
- configuration file, STS
- configuration, base configuration parameter 77
- configuration, installation, and distribution (CID)
 - installation method 27
- connection configuration file, SNA/DS
 - editing, SNA/DS 107
 - editing, STS 98
 - example for SNA/DS with APPC 109
 - example for STS with TCP/IP 99
 - next DSU 108
 - parameters 107
 - protocol, SNA 107
 - protocol, STS 99
 - receive MU_ID time-out 109
 - receive TP symbolic destination 108
 - retry limit 109
 - send MU_ID time-out 109
 - send TP symbolic destination 108
 - server name, SNA 108
 - server name, STS 99
 - TCP/IP time-out 108
 - transmission time-out 109
- connection configuration file, STS
- connection window duration, base configuration
 - parameter 78
- connectivity 22
- corequisites, in software object 25
- CRC numbers 266
- cx_check_crcs 266

cx_check_secure_key 266
 cx_daca_filename 257
 cx_daca_report 257
 cx_daca_request 258
 cx_mount_fs 259
 cx_set_secure_key 265
 cx_unmount_fs 260
 cx_vercm 263

D

DACA idle time, base configuration parameter 78
 DACA retry time, base configuration parameter 78
 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 202
 database recovery tool 201
 DEFAULT.SID inventory dictionary file 243
 defining user authorization profiles 39
 delete pending request xxv
 dictionary file, inventory discovery 244
 directory specifications in software object 25
 discovered, software object status 36
 discovery, inventory
 dictionary file 244
 hardware filter 247
 running 247
 supplied program 243
 disk requirements
 requirements for TME 10 Software Distribution
 components 49
 disk space specifications in software object 25
 DiskCamera
 drive for 227
 files monitored by 227
 scenario 227
 workplace objects monitored by 227
 DiskPicture
 basic scenario 229
 distributed, software object status 36
 distribution pending, software object status 36
 distribution server
 attended installation 53

distribution server (*continued*)
 hardware requirements 50
 installation requirements 49
 unattended installation 70
 uninstalling 70
 upgrading 66
 DLLs for user exits 268
 documentation folder 44
 domain 4
 drive
 for DiskCamera 227
 DSKCAM.INI file 227
 dynamic groups 23
 dynamic software object
 example of creation 217
 purpose 28
 dynamic software objects 28

E

EBCDIC files, translation 34
 editing remote connection files 97, 105
 end user 10
 end user authorization profile 41
 environment variables
 ACT_ON_TARGET 269
 FNDBOOT 271
 FNDHIDE 271
 FNDLOGMSEC 270
 FNDNOOK 271
 FNDNTGUI 271
 FNDPASSWORD 270
 FNDREQUESTS 269
 FNDUEX_TRC 270
 FNDUSER 270
 GI_RETRY_TIME 270
 HIDE_HIST_CM_PB 270
 STS_PERM_CONNECTION 271
 examples
 advanced software preparation 211
 basic software preparation 202
 discovered software package 245
 DiskCamera 227
 dynamic software object 217
 quiesce script 249
 remote directory 211
 exit codes, scripts 253
 exporting directories, user exit 258

Index

F

- fan out requests 20
- file system, base configuration parameter 78
- files provided for user exits 255
- filter, hardware inventory discovery 247
- filters for dynamic groups 23
- fix software object 26
- FNDACTREQUESTS 269
- FNDADMN 39
- FNDBLD 40
- FNDBOOT 271
- FNDHIDE 271
- FNDHWINV hardware inventory file 246
- FNDINV inventory discovery program 243
- FNDITRC 269
- FNDITRC_CO 269
- FNDITRC_DAS 269
- FNDITRC_GI 269
 - refid=env.FNDITRC 269
 - refid=env.FNDITRC_CO 269
 - refid=env.FNDITRC_DAS 269
 - refid=env.FNDITRC_GI 269
- FNDLOGMSEC 270
- FNDNOOK 271
- FNDNTGUI 271
- FNDPASSWORD 270
- FNDSWINV software inventory file 245
- FNDTKINV installation parameters file 245
- FNDUEX_TRC 270
- FNDUSER 41, 270
- focal mode 90
- fully disconnected client 19

G

- generic installation method 27
- GI_RETRY_TIME 270
- global name
 - See label
- grouping targets 23

H

- hardware inventory discovery 246
- hardware prerequisites
 - in software object 25
- header files, user exits 256
- help window 11

- help, online 44
- HIDE_HIST_CM_PB 270
- holding operations 33
- host deletion of pending requests xxv

I

- IBM service, calling 46
- in error, software object status 36
- in progress, software object status 36
- inactive, software object status 36
- INF files
 - 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 Windows NT publications 44
- information folder 44
- installation, TME 10 Software Distribution
 - planning 49
 - response file, Windows NT or Windows 2000 70
- installed, software object status 36
- interface
 - command line, description 13
 - graphical user, description 10
 - GUI 10
 - starting, graphical 14
- intermediate node 20
- inventory discovery
 - dictionary file 244
 - hardware filter 247
 - running 247
 - supplied program 243
 - writing an inventory program 244
- inventory program, base configuration parameter 78
- IPX
 - customizing a Windows NT or Windows 2000 client for 65
 - customizing a Windows NT or Windows 2000 server for 64

L

- label 26
- LAN Adapter and Protocol Support (LAPS)
- LAN authorization, base configuration parameter 79

LANTRAN.LOG
 LAPS (LAN Adapter and Protocol Support)
 local port, TCP/IP 85
 local queues window 11
 local target statuses, changes to xxvi
 local targets 17
 log file size, base configuration parameter 79
 LU name 100, 111

M

machine type, base configuration parameter 79
 make files for user exits 268
 manager mode 90
 manuals
 INF files 44
 PDF files 46
 PostScript 45
 TME 10 Software Distribution products xxi
 TME 10 Software Distribution, Version 3.1.5 for
 Windows NT 44
 max attempts, base configuration parameter 79
 max connections, base configuration parameter 79
 max CRMU, SNA/DS configuration file 106
 max local targets, base configuration parameter 79
 max requests, base configuration parameter 80
 max server conns, base configuration parameter 80
 max server targets, base configuration parameter 80
 max SNADS routes, base configuration parameter 80
 max STS routes, base configuration parameter 80
 max STS, base configuration parameter 80
 max targets, base configuration parameter 81
 max user interfaces, base configuration file
 parameter 81
 message help 44
 message log interface 11
 message log level, base configuration parameter 81
 message log window 11
 message reference online file 44
 migration scenarios 44
 minimum distribution priority 113
 Communications Server,
 configuring 145
 MS SNA Server
 configuring 117
 Personal Communications
 configuring 173
 mobile client 18
 modes, target
 client 17

 modes, target (*continued*)
 mobile client 18
 server 17
 user interface only 19
 modifying local file names, user exit 257
 mounting file systems, user exit 259
 MPTS (Multi Protocol Transport Support)
 MS SNA Server
 configuring 117
 Multi Protocol Transport Support (MPTS)

N

name, of target 91
 name, server 86
 NetBIOS
 customizing a Windows NT or Windows 2000 client
 for 64
 customizing a Windows NT or Windows 2000 server
 for 61
 NetFinity Manager, requirement for inventory
 discovery 243
 network configurations
 clients, definition 4
 domain, definition 4
 TME 10 Software Distribution in 3
 network ID 100, 111
 new in this release xxiii
 next DSU, SNA/DS connection configuration file 108
 not authorized, software object status 36
 not installed, software object status 36
 not removable, software object status 36
 notifying completion, user exit 257
 nvdm.cfg
 See base configuration file

O

origin hop count, SNA/DS configuration file 106
 origin hop count, STS configuration file 98
 OS/2 client
 files monitored by DiskCamera 227
 supported 21
 OS2.INI
 monitoring in a DiskCamera installation 227
 OS2INIT.CMD
 monitoring in a DiskCamera installation 227
 OS2SYS.INI
 monitoring in a DiskCamera installation 227

Index

overview 3

P

passive client 5
PDF format books 46
pending requests, delete xxv
Personal Communications
platforms supported xxiii
platforms, client 21
platforms, remote 21
ports, TCP/IP 85
post-accept scripts 252
post-install scripts 251
post-remove scripts 252
post-uninstall scripts 253
PostScript publications 45
pre-accept scripts 252
pre-activate scripts 253
pre-install scripts 251
pre-remove scripts 252
pre-uninstall scripts 253
preparation site 20
preparation, software
 advanced example 211
 basic example 202
 examples, DiskCamera 227
prerequisites, hardware and software
 host communication 51
 software, server 50
prerequisites, in software object 25
previewing and modifying reports, user exit 257
pristine installation 34
pristine installation scenarios, new xxiii
pristine scenarios 44
processor requirements
 server 49
products, in software object 25
profiles, user authorization 39
programs, in software object 25
protocol, base configuration parameter 81
protocol, SNA/DS connection configuration file 107
protocol, STS connection configuration file 99
protocols
 customizing NVDM.CFG for NetBIOS and IPX 61
 supported 22
 supported by all Windows Operating Systems 50
publications
 INF files 44
 PDF files 46

publications (*continued*)
 PostScript 45
 TME 10 Software Distribution products xxi
 TME 10 Software Distribution, Version 3.1.5 for
 Windows NT 44
pull mode 90
 definition 17
push mode 90
 definition 17

Q

quiesce check 249

R

RAM requirements
 server 49
READ.ME file 44
reboot required, software object status 36
receive MU_ID time-out, SNA/DS connection
 configuration file 109
receive TP symbolic destination, SNA/DS connection
 configuration file 108
reconfiguring
recovery tool 201
recursive operations, scheduling 33
refresh software object 26
Registry
 monitoring in a DiskCamera installation 227
remote directory, example 211
remote platforms supported 21
remote port, TCP/IP 85
remote queues window 11
remote source software object 25
remote targets 19
removable, software object status 36
removing directories, user exit 259
removing settings 72
reporting fileset status and products, user exit 263
reporting request completion, user exit 263
repository, base configuration parameter 83
requests window 11
requirements
 host communication 51
 software, server 50
response files
 in software object 25
 installing Windows NT or Windows 2000 server
 with 70

retry limit, SNA/DS connection configuration file 109
 root user
 security 39

S

samples

advanced software preparation 211
 basic software preparation 202
 discovered software package 245
 DiskCamera 227
 dynamic software object 217
 quiesce script 249
 remote directory 211

scheduling facilities

change control operations 32

scripts

exit codes 253
 in software object 25
 post-accept 252
 post-install 251
 post-remove 252
 post-uninstall 253
 pre-accept 252
 pre-activate scripts 253
 pre-install 251
 pre-remove 252
 pre-uninstall 253
 writing 251

SD4DOCPS directory 45

secure key 42

secure package 42

security 39

send MU_ID time-out, SNA/DS connection configuration file 109

send TP symbolic destination, SNA/DS connection configuration file 108

server name 100, 111

server name, SNA/DS connection configuration file 99, 108

server target 19

server, base configuration parameter 83

server, distribution

hardware prerequisites 49
 hardware requirements 50
 installation 53
 installation with response file 70
 publications xxi
 software prerequisites 50
 uninstallation 70

server, distribution (*continued*)

upgrade 66

server, target type 92

service area, base configuration parameter 83

service, IBM 46

setting secure keys 265

settings, removing 72

single node

as a preparation site 9

purposes 20

single, target type 92

SNA Server

See MS SNA Server

SNA/DS address 111

SNA/DS configuration file

editing remote connection files 105

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 245

software object

allocation failure retry time 106

CID 27

compression 25

contents of 25

corequisites 25

description 25

directory specifications in 25

disk space specification 25

dynamic 28

editing, SNA/DS 105

editing, STS 97

example, SNA/DS 106

example, STS 98

fix 26

generic 27

labels 26

max CRMU 106

naming 26

origin hop count, SNA 106

origin hop count, STS 98

parameters, SNA 105

parameters, STS 98

platform specific 26

prerequisites 25

products in 25

Index

- software object (*continued*)
 - programs in 25
 - refresh 26
 - remote source 25
 - response files in 25
 - scripts in 25
 - security 42
 - status 36
 - transmission hold time 106
 - update 26
 - software object profiles window 12
 - software preparation
 - advanced example 211
 - basic example 202
 - examples, DiskCamera 227
 - software preparation interface 12
 - software prerequisites
 - host communication 51
 - in software object 25
 - server 50
 - Software profile - create another 12
 - specifications, software object 25
 - SPX
 - ss_user_loc_name 256
 - starting graphical interface 14
 - starting the command line interface 13
 - starting TME 10 Software Distribution 13
 - STARTUP.CMD
 - monitoring in a DiskCamera installation 227
 - static software objects 28
 - stattg command, changes to statuses xxvi
 - stopping the command line interface 13
 - stopping TME 10 Software Distribution 15
 - STS configuration file
 - description 97
 - STS connection configuration file
 - description 98
 - STS idle time, base configuration parameter 83
 - STS retry time, base configuration parameter 83
 - STS routing table
 - description 99
 - STS_PERM_CONNECTION 271
 - sx_export_dir 258
 - sx_server_report
 - notifying completion 257
 - reporting request completion 263
 - sx_server_request 256
 - sx_unexport_dir 259
 - sx_usrexit_trgcfg 260
 - sx_usrexit_usrcfg 262
- ## T
- TAK 41
 - target address 100, 111
 - target password authentication, base configuration parameter 84
 - targets
 - access keys 41
 - automatic registration 20
 - dynamic groups 23
 - grouping 23
 - holding operations at 33
 - intermediate node 20
 - local 5
 - maximum concurrently active 24
 - pristine installations on 34
 - remote 5
 - server 19
 - single node 20
 - user access to 41
 - targets window 11
 - TCP/IP
 - TCP/IP port, base configuration parameter 84
 - TCP/IP time-out, SNA/DS connection configuration file 108
 - trace file size, base configuration parameter 84
 - translation, data 34
 - transmission hold time, SNA/DS configuration file 106
 - transmission protocols
 - supported 22
 - transmission time-out, SNA/DS connection configuration file 109
 - type, target 92
- ## U
- UI only targets 93
 - unattended installation
 - overview 32
 - TME 10 Software Distribution server 70
 - uninstallation
 - TME 10 Software Distribution server 70
 - unmounting file systems, user exit 260
 - update software object 26
 - upgrading
 - TME 10 Software Distribution server 66
 - user exits 255
 - adding target parameters 260

user exits (*continued*)

- adding user parameters 262
- checking CRC numbers 266
- checking secure keys 266
- compilation methods 268
- cx_check_crcs 266
- cx_check_secure_key 266
- cx_daca_filename 267
- cx_daca_report 257
- cx_daca_request 258
- cx_mount_fs 259
- cx_set_secure_key 265
- cx_unmount_fs 260
- cx_vercm 263
- DLLs 268
- exporting directories 258
- functions supported by 255
- generating local file names 256
- header files 256
- make files 268
- mounting file systems 259
- notifying requests 256
- previewing and modifying reports 257
- previewing requests, user exit 258
- removing exports 259
- reporting fileset status and products 263
- reporting request completion 263
- setting secure keys 265
- ss_user_loc_name 256
- sx_export_dir 258
- sx_server_report, notifying completion 257
- sx_server_report, request completion 263
- sx_server_request 256
- sx_unexport_dir 259
- sx_usrexit_trgcfg 260
- sx_usrexit_usrcfg 262
- unmounting file systems 260
- writing 255

user interface only client 19

user interface only targets 93

users

- administrator 39
- authorization profiles 39
- authorizations 89
- builder 9
- builders 40
- change control administrator 9
- data access keys 41
- defining 39
- end user 10

users (*continued*)

- root 39
- target access keys 41
- users belonging to FNDUSER 41

V

variables

- customizing 236

VTAM

W

what's new xxiii

Windows 2000 clients

Windows 3.1 clients

- files monitored by DiskCamera 227
- supported 21

Windows 95 clients

- files monitored by DiskCamera 227
- supported 21

Windows 98 clients

Windows NT clients

- files monitored by DiskCamera 227
- supported 21

windows, graphical user interface

- catalog 10

- help 11

- local queues 11

- message log 11

- requests 11

- software object profiles 12

- Software profile - create another 12

- targets 11

work area, base configuration parameter 84

workplace objects, monitored by DiskCamera 227

workstation name, base configuration parameter 84

writing change control scripts 251

X

XVIEW command 44

Communicating Your Comments to IBM

TME® 10 Software Distribution
for Windows NT®
Quick Beginnings
Version 3.1.5
Publication No. SH19-4335-02

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 Windows NT®
Quick Beginnings
Version 3.1.5**

Publication No. SH19-4335-02

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:

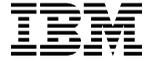
Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Help us help you!
SH19-4335-02



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

SH19-4335-02

Cut or Fold
Along Line



Program Number: 5639-B01

Distributed Electronically for Customer Printing