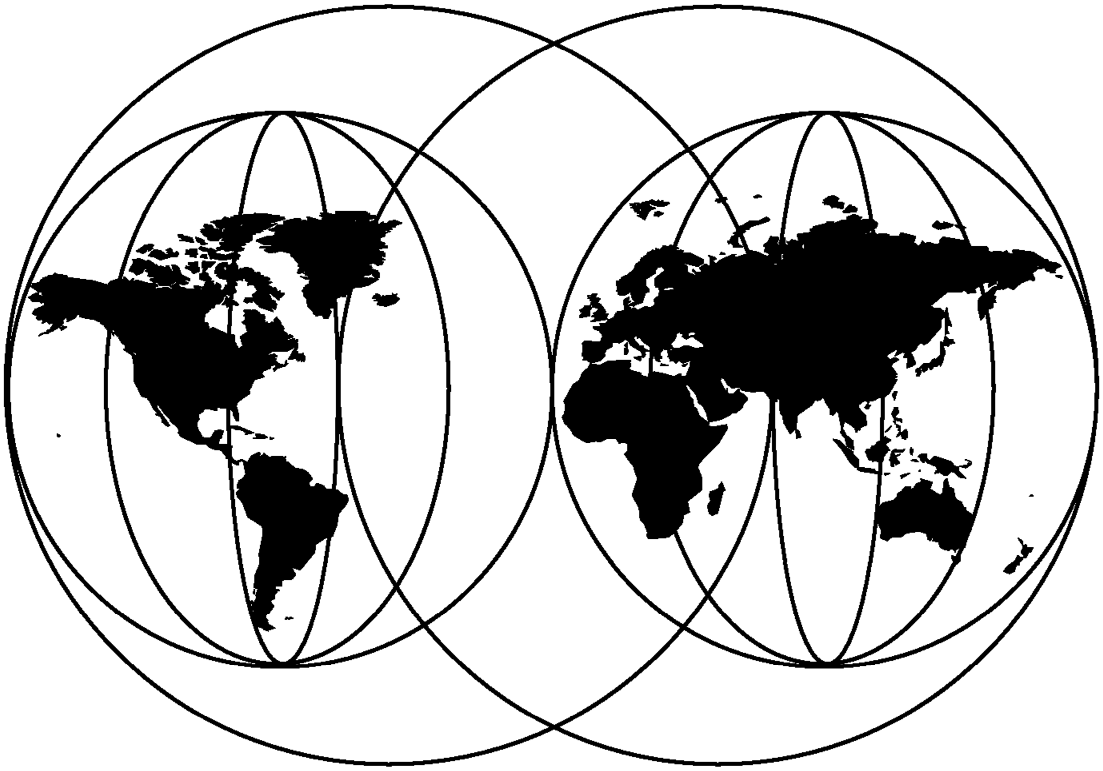
Data Sharing: Implementing InfoSpeed Copy-Sharing Solutions Among Heterogeneous Systems

Cyril Anam, Marc Chapel, Keijo Ekman, Marty Gardner,
Kjell E Nystrom



International Technical Support Organization

http://www.redbooks.ibm.com

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see “How to Get ITSO Redbooks” at the back of this book for ordering instructions.

SG24-4120-00

International Technical Support Organization

SG24-4120-00

Data Sharing: Implementing InfoSpeed Copy-Sharing Solutions Among Heterogeneous Systems

December 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix G, “Special Notices” on page 219.

First Edition (December 1998)

This edition applies to Version 1, Release 1 of IBM InfoSpeed exPress Data Mover, Programs 5697-D61 and 5648-PDM made general available in June 1998.

Note

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Figures ix

Tables xiii

Preface xv
The Team That Wrote This Redbook xv
Comments Welcome xvi

Chapter 1. Introduction to Data Sharing 1
1.1 Types of Data Sharing 2
1.1.1 Storage Sharing 2
1.1.2 Data-Copy Sharing 3
1.1.3 True Data Sharing 4
1.2 Business Values of the IBM InfoSpeed Solution 6
1.2.1 InfoSpeed and Data Currency 8
1.2.2 InfoSpeed Information Sharing 8
1.2.3 InfoSpeed and Storage Management 8
1.2.4 InfoSpeed and Data Mining and Data Warehousing 9
1.2.5 InfoSpeed and High-Volume Printing 9
1.2.6 InfoSpeed and Use of Tape Library 9
1.2.7 Tape Management System for UNIX/NT 9
1.2.8 InfoSpeed and ADSM 10
1.2.9 InfoSpeed and Backup Solutions 10
1.2.10 Create an Information Network 10
1.2.11 InfoSpeed and Data Archive Solutions 10
1.2.12 InfoSpeed Benefit Summary 11

Chapter 2. IBM InfoSpeed Technical Overview 13
2.1 IBM InfoSpeed Data Gateway 14
2.2 IBM InfoSpeed Multiplexor 15
2.3 IBM exPress Data Mover 15
2.4 Hardware and Software Requirements 16
2.4.1 S/390 Server 16
2.4.2 RS/6000 and SP 17
2.4.3 Hewlett-Packard 9000-800 17
2.4.4 Sun Microsystems Ultra Enterprise 17
2.4.5 PC Servers 17

Chapter 3. IBM InfoSpeed: Configuration Alternatives 19
3.1 Direct ESCON Channel Attachment 19
3.2 IBM InfoSpeed Data Gateway Attachment 20
3.3 Adding the IBM InfoSpeed Multiplexor 22
3.4 Adding PDM Network-Attached Nodes 23
3.5 Adding Non-PDM Nodes 24

Chapter 4. IBM InfoSpeed: Concepts and Principles 25
4.1 The PDM MVS Subsystem 25
4.2 The PDM UNIX or Windows NT Subsystem 25
4.3 User Interfaces 26
4.3.1 PDM File (MVS or Node) Transfer Client 26
4.3.2 PDM ISPF Client 27

4.3.3 PDM Subsystem Interface 27
4.3.4 PDM REXX Interface 27
4.4 The IBM exPress Data Mover Architecture 27
4.4.1 SCSI Target IDs/LUN#s 28
4.4.2 ESCON Subchannels 29
4.4.3 Multiple Connections to a Single Server 29
4.5 The IBM exPress Data Mover and Its Paths on OS/390 29
4.6 The IBM exPress Data Mover Used on UNIX or Windows NT 30
4.6.1 PDM Nodes 30
4.6.2 Non-PDM Node 30

Chapter 5. Install and Configure IBM InfoSpeed 31
5.1 IBM InfoSpeed Data Gateway installation 31
5.2 IBM InfoSpeed Multiplexor Installation 32
5.3 IBM InfoSpeed Ordering Information 33
5.4 IBM InfoSpeed Configuration 34

Chapter 6. PDM MVS Installation and Configuration 43
6.1 MVS Software Prerequisites 43
6.2 PDM MVS Subsystem Installation Steps 43
6.2.1 Define Devices to MVS IOCP or to HCD 44
6.2.2 Disable Missing Interrupt Handler 45
6.2.3 Define PDM as a Subsystem 45
6.2.4 Load the Required Files from the PDM Distribution Tape 45
6.2.5 Install the Distribution Tape 47
6.2.6 Install the Cumulative Service Tape 47
6.2.7 Allocate the PDM Data Sets 47
6.2.8 Adding DDDEFS to SMP/E 47
6.2.9 Define SDMELINK as an Authorized Library 47
6.2.10 Add SDMELPA to the LPALST 48
6.2.11 Add SDMELOAD to the LNKLST 48
6.2.12 Edit Start Procedure 48
6.2.13 Concatenate SDMEPROC to the PROCLIB list 48
6.2.14 Initialize the PDM Queue 48
6.2.15 Define Nodes to PDM 48
6.2.16 Specifying Data Set Security 49
6.2.17 Optional Steps 50
6.3 PDM MVS Subsystem Configuration 51
6.3.1 The DMETABLE Program 51
6.3.2 HDETABLE Initialization Sample 55
6.4 PDM MVS Subsystem Installation Checklist 60
6.5 PDM MVS Subsystem Operation 60
6.5.1 Starting and Stopping the MVS DMES Subsystem 61
6.5.2 Verifying MVS and Remote Node Connectivity 62

Chapter 7. Install and Configure PDM Channel-Attached Nodes 65
7.1 AIX Installation Planning 65
7.1.1 Install the IBM exPress Data Mover 65
7.1.2 Configure the ESCON Adapter 65
7.1.3 Configure the SCSI Adapter 71
7.1.4 Configure IBM exPress Data Mover 72
7.2 HP/UX Installation Planning 76
7.2.1 Prerequisites 76
7.2.2 Install the IBM exPress Data Mover 76
7.2.3 SCSI Adapter I/O Definition 77

7.2.4 Configure IBM exPress Data Mover 78
7.3 Solaris Installation Planning 81
7.3.1 Prerequisites 81
7.3.2 Software Installation 81
7.3.3 SCSI Adapter I/O Definition 82
7.3.4 Configure IBM exPress Data Mover 83
7.4 Windows NT Installation Planning 87
7.4.1 Prerequisites 87
7.4.2 Windows NT SCSI Adapter I/O Definition 87
7.4.3 Install the IBM exPress Data Mover 90
7.4.4 Configure IBM exPress Data Mover for Windows NT 93

Chapter 8. Install and Configure PDM Network-Attached or Downstream Nodes 97
8.1 UNIX Downstream Nodes 97
8.2 Windows NT Downstream Nodes 100

Chapter 9. Configure a PDM FTP Server for Non-PDM Nodes 103
9.1 PDM FTP Server on UNIX 103
9.1.1 Connection to One OS/390 Host 103
9.1.2 Connection to Multiple OS/390 Servers 104
9.2 PDM FTP Server on Windows NT 104
9.3 Non-PDM Node 104

Chapter 10. Using MVS PDM Clients 105
10.1 PDM MVS Client 105
10.1.1 Using the PDM MVS File Transfer Client (DMEFTP) 106
10.1.2 Using the MVS ISPF Client (DMESISPF) 111
10.1.3 Using the PDM REXX Interface 123
10.2 Using the MVS PDM Subsystem Interface 124
10.2.1 Activating the SSI 124
10.2.2 Running an SSI Job 124
10.3 MVS Data Sets 127
10.3.1 PDS and PDSE Specifications 129
10.3.2 Using Model Data Set Control Blocks 130
10.3.3 Using Generation Data Groups 130
10.3.4 Virtual Storage Access Method (VSAM) Data Sets 131
10.3.5 Using Tapes 131
10.3.6 Migrated Data Sets 131
10.3.7 Directory Emulation 131
10.4 Piping Data Using GETX/PUTX Commands 132
10.4.1 How to Pipe Data on MVS 133

Chapter 11. Performance Expectations 135
11.1 Performance Expectations Using the Data Gateway 135
11.2 Performance Expectations for ESCON Only 136
11.3 Performance Considerations 136
11.4 Performance in Terms of Data Transfer Rates 137
11.5 Performance in Terms of Elapsed Time 138
11.6 Increasing Bandwidth or Parallelism for a Server 139

Chapter 12. Data Copy Sharing Samples 141
12.1 Connecting Options 141
12.2 Data-Copy Sharing Samples Using MVS 141
12.2.1 Using the File Transfer Client 142

12.2.2 Using The PDM Interactive TSO Client 149
12.2.3 Using JCL and the SSI 150
12.3 Backing up Data to S/390 Environment 157
12.3.1 Backing up Databases from UNIX or Windows NT to OS/390 . . . 157
12.3.2 Backing up Data from UNIX to S/390 Environment 158
12.4 Data-Copy Sharing Samples from the UNIX or Windows NT 160
12.4.1 Get Data from MVS 160
12.4.2 Put Data on MVS 161
12.4.3 Set up Default Account Information 163
12.4.4 Using Default Allocation Information on OS/390 163
12.4.5 Submitting an Asynchronous Request 164
12.4.6 Copying Several Binary Files 165
12.4.7 Submitting Multiple Requests 166

Appendix A. Configuration Worksheets 167
A.1 InfoSpeed Gateway Configuration 167
A.1.1 Gateway Configuration Chart 167
A.1.2 Gateway Configuration Worksheets 171
A.1.3 Direct ESCON Attachment Configuration Worksheets 179

Appendix B. OS/390 Security Processing in IBM exPress Data Mover 185
B.1 PDM Security Before APAR OW34962 185
B.2 PDM Security After APAR OW34962 185

Appendix C. Small Computer Systems Interface (SCSI) Tutorial 187
C.1 Interfaces 187
C.2 Data Transfer 188
C.3 Configurations 189
C.4 I/O Process 190
C.5 Addressing 190
C.6 Tape I/O Commands 191
C.7 Bus Options 191
C.8 Other Facilities 193
C.9 Performance 193
C.10 Connecting SCSI Devices 193
C.11 Differences between SCSI-1 and SCSI-2 195
C.12 Differences between SCSI-2 and SCSI-3 196
C.13 Command Codes for Sequential-Access Devices 197
C.14 Command Codes for Medium Changer Devices 198

Appendix D. A Short Introduction to DFSMS 201
D.1 Data Sets 201
D.1.1 Naming Data Sets 201
D.1.2 Cataloging Data Set Names 201
D.2 Using DFSMS 202
D.2.1 The DFSMS Environment 202
D.2.2 Benefits of System-Managed Storage 203
D.2.3 Managing Data with SMS 205

Appendix E. UNIX basics 211
E.1 Login to the system 211
E.2 Go to the Proper Directory 211
E.3 Launch Interactive Management Menus 212
E.4 Use the Editor 212
E.4.1 How to Use vi 213

Appendix F. Very Short FTP Introduction 217

Appendix G. Special Notices 219

Appendix H. Related Publications 223
H.1 International Technical Support Organization Publications 223
H.2 Redbooks on CD-ROMs 223
H.3 Other Publications 223

How to Get ITSO Redbooks 225
IBM Redbook Fax Order Form 226

List of Abbreviations 227

Index 229

ITSO Redbook Evaluation 231

Figures

1. Data Sharing Concept and Types . . . 2
2. Example Showing How Storage Sharing Works . . . 3
3. Data-Copy Sharing Using a Three-Step Process . . . 4
4. Data-Copy Sharing Using a One-Step Process . . . 4
5. True Data Sharing . . . 5
6. True Data Sharing in Homogeneous Environments . . . 5
7. IBM InfoSpeed Solution . . . 6
8. InfoSpeed Connection Options . . . 7
9. Components of an IBM InfoSpeed Solution . . . 13
10. IBM InfoSpeed Data Gateway Model DG1 . . . 14
11. IBM InfoSpeed Multiplexor Model MX1 . . . 15
12. Configuration with the IBM exPress Data Mover . . . 16
13. IBM InfoSpeed Basic Configuration . . . 19
14. IBM InfoSpeed Configuration with One IBM InfoSpeed Data Gateway . . . 20
15. Different Configurations Using One or More IBM InfoSpeed Data Gateways. . . . 21
16. Full IBM InfoSpeed Configuration . . . 22
17. Adding PDM Network-Attached Nodes to a Basic Configuration . . . 23
18. Adding Non-PDM Nodes . . . 24
19. PDM User Interface Examples . . . 26
20. IBM InfoSpeed Data Gateway Connections . . . 31
21. IBM InfoSpeed Multiplexor Connections . . . 32
22. IBM InfoSpeed Data Gateway and IBM InfoSpeed Multiplexor Cabling and Features . . . 33
23. Sample InfoSpeed Installation Project . . . 34
24. Sample InfoSpeed Configuration Chart . . . 35
25. Sample JCL to Copy Install Members from PDM Distribution Tape . . . 46
26. Sample JCL to Configure the HDETABLE. . . . 56
27. Sample MVS PDM Configuration . . . 59
28. PDM LOG Messages Indicating a Successful DMESYS Startup and Connection . . . 63
29. SMIT System Management Screen . . . 66
30. SMIT Devices Screen . . . 66
31. SMIT Communication Screen . . . 67
32. SMIT ESCON Channel Adapter Screen . . . 67
33. SMIT Subchannel Definitions Screen . . . 67
34. ESCON Subchannel Configuration Menus in SMIT . . . 68
35. SMIT Subchannels Definitions . . . 68
36. ESCON Subchannel Characteristics . . . 69
37. SMIT ESCON Channel Adapter Menu . . . 69
38. ESCON Subchannel Characteristics . . . 70
39. SMIT ESCON Channel Adapter Menu . . . 70
40. SMIT Fiber Definitions . . . 70
41. Add a Fiber Dialog Screen . . . 71
42. List of Tape Devices . . . 71
43. PDM Installation Menu on AIX(1) . . . 72
44. PDM Installation Menu on AIX(2) . . . 73
45. PDM Installation Menu on AIX(3) . . . 73
46. PDM Installation Menu on AIX(4) . . . 73
47. IBM exPress Data Mover Installation Steps on AIX . . . 73
48. IBM exPress Data Mover: End of Installation on AIX . . . 75

49. PDM Installation Menus on HP(1) 78
50. PDM Installation Menus on HP(2) 79
51. PDM Installation Menus on HP(3) 79
52. PDM Installation Menus on HP(4) 79
53. PDM Configuration File Completed for HP/UX 79
54. IBM exPress Data Mover: End of Installation on HP/UX 81
55. PDM Installation Menu on Sun(1) 84
56. PDM Installation Menu on Sun(2) 84
57. PDM Installation Menu on Sun(3) 84
58. PDM Installation Menu on Sun(4) 85
59. PDM Configuration File Completed for Sun Solaris 85
60. IBM exPress Data Mover: End of Installation on Sun Solaris 86
61. SCSI Channel Definition on Windows NT (1) 88
62. SCSI Channel Definition on Windows NT (2) 88
63. SCSI Channel Definition on Windows NT (3) 89
64. SCSI Channel Definition on Windows NT (4) 89
65. PDM Installation Menu on Windows NT (1) 90
66. PDM Installation Menu on Windows NT (2) 90
67. PDM Installation Menu on Windows NT (3) 91
68. PDM Installation Menu on Windows NT (4) 91
69. PDM Installation Menu on Windows NT (5) 92
70. PDM Installation Menu on Windows NT (6) 92
71. PDM Installation Menu on Windows NT (7) 93
72. Configuration File Completed for Windows NT (8) 93
73. PDM Configuration Menu on Windows NT (1) 94
74. PDM Configuration Menu on Windows NT (2) 94
75. PDM Configuration Menu on Windows NT (3) 95
76. PDM Configuration Menu on Windows NT (4) 95
77. PDM Configuration Menu on Windows NT (5) 96
78. PDM Configuration Menu on an UNIX Downstream Node (1) 97
79. PDM Configuration Menu on an UNIX Downstream Node (2) 98
80. PDM Configuration Menu on an UNIX Downstream Node (3) 98
81. PDM Configuration Menu on an UNIX Downstream Node (4) 98
82. PDM Configuration File Complete for an UNIX Downstream Node 98
83. PDM Installation Complete for a UNIX PDM Network-Attached Node . 100
84. PDM Configuration Menu on Windows NT Downstream Node (1) . . . 101
85. PDM Configuration Menu on Windows NT Downstream Node (2) . . . 101
86. PDM Configuration Menu on Windows NT Downstream Node (3) . . . 102
87. The Configuration Used in the MVS PDM Client Examples 105
88. Sample JCL for PDM MVS File Transfer Client GET Command 107
89. PDM MVS File Transfer Client Commands for an Interactive TSO Session 108
90. Command Library JCL Sample for AUTOLOG 109
91. Command Library JCL Sample to Execute Multiple Members 110
92. Using Command Library in Interactive Session 110
93. Sample Code for Using Command Library in Interactive Session (Multiple Members) 111
94. PDM MVS File Transfer Client Main Panel 112
95. PDM MVS File Transfer Client Panel, Download Example 113
96. PDM MVS File Transfer Client Panel, Download Response Example . 114
97. PDM MVS File Transfer Client Event Manager Panel 115
98. Event Manager Status Panel 116
99. Event Manager Detail Panel 118
100. Event Manager Cancel Panel 118
101. PDM MVS File Transfer Client Transfer Path Monitor panel. 119

102. PDM MVS File Transfer Client CTASK Detail Panel 121
103. Stopping a Communications Task from the PDM MVS File Transfer Client CTASK Detail Panel. 122
104. PDM MVS File Transfer Client Monitor Panel 122
105. Sending Data from UNIX to UNIX Using SSI 126
106. Piping Data Using GETX Command 133
107. Piping Data To Database Loader 133
108. Maximum Data Transfer Rates for Selected Server Interconnect Options 137
109. Minimum Time Needed to Restore 50 2-GB Volumes for Selected . . . 138
110. Example of an Increased Bandwidth Configuration 139
111. Device Number Distribution 139
112. Connecting Options for Data-Copy Sharing 141
113. Sample JCL for PDM MVS File Transfer Client Using GET Command with Dataclass 142
114. Sample JCL for PDM MVS File Transfer Client PUT command 143
115. Sample JCL for PDM MVS File Transfer Client Copy from UNIX to a MVS Tape 144
116. Sample JCL for PDM MVS File Transfer Client Copying Multiple Files from UNIX 145
117. Sample JCL For PDM MVS File Transfer Client Copying Multiple Members to UNIX 146
118. Sample JCL for PDM MVS File Transfer Client to Write an MVS Data Set from a UNIX Application 147
119. Sample JCL For PDM MVS File Transfer Client To Read a MVS Data Set From a UNIX Application 148
120. Using the MVS ISPF Client Interactive Session 149
121. SSI-JCL to Copy a UNIX or Windows NT File to MVS 151
122. SSI-JCL to Copy MVS Data Sets to UNIX or Windows NT 152
123. SSI-JCL to Read a UNIX or Windows NT File 153
124. SSI-JCL to Write a UNIX or Windows NT File 153
125. SSI-JCL to Pipe Data from a UNIX Executable 154
126. SSI-JCL to Pipe Data to a UNIX Executable 154
127. SSI-JCL to Pipe Data Between an MVS Application and UNIX Executables 155
128. SSI PROFILE Sample 155
129. SSI-JCL to Copy UNIX or Windows NT to UNIX or Windows NT 156
130. Submit MVS-JCL from a PDM Client 156
131. SSI-JCL For OpenEdition 156
132. SSI-JCL to Pipe Data Between UNIX Executables 157
133. Backing up UNIX/NT Data Base to S/390 Tape Library 158
134. Backing up Data from UNIX to S/390 Environment 158
135. Example of JCL to Back up UNIX Data Using tar 159
136. SSI-JCL to Back up an HP Raw Partition 159
137. Sample Script to Get Data from MVS 160
138. Sample Script to Put Data on MVS 162
139. Script to Set up Default Account Information in PDM 163
140. Script Defining a Source File for PDM 163
141. Script to Start DMEFTP Using a Profile Source File 164
142. Exampl Script for an Asynchronous Request 165
143. Example of Script to Copy Several Binary Files 165
144. Sequence of Instructions to Submit Multiple Requests 166
145. Gateway Configuration Worksheet - Planning Chart 170
146. SCSI Multiple Initiators and Multiple Targets 189
147. SCSI I/O Operation Sequence 190

148. Products that Make Up the DFSMS Environment 203
149. Diagram Showing Processing of ACS Routines 210

Tables

1. Information Technology Values and Business Values of IBM InfoSpeed 7
2. ESCON Port 1 MVS Host Information; Customer name: TUCSON RESIDENCY 37
3. SCSI Information for ESCON Port 1; Customer name: TUCSON RESIDENCY 38
4. (* pisces) Direct ESCON Adapter UNIX Information; Customer name: TUCSON RESIDENCY 39
5. (* pisces) Direct ESCON Adapter (Part 1 of _1) S/390 Information; Customer name: TUCSON RESIDENCY 40
6. DMETABLE Parameter Types and Explanation of Each Parameter 52
7. DMETABLE SDEF Statement Parameter Names and Explanation 54
8. PDM Event Manager Panel Options 115
9. PDM Event Manager Status Panel Fields 117
10. PDM Event Manager Monitor Panel Fields 120
11. PDM SITE and LSITE Parameters 128
12. InfoSpeed Performance 135
13. PDM ESCON Only Performance 136
14. Required Customer Information 172
15. ESCON Port 1 MVS Host Information; Customer name:____ . . . 173
16. SCSI Information for ESCON Port 1; Customer name:____ . . . 174
17. ESCON Port 2 MVS Host Information; Customer name:____ . . . 175
18. SCSI Information for ESCON Port 2; Customer name:____ . . . 176
19. (*_____________) Direct ESCON Adapter UNIX Information; Customer name:____________________ 180
20. (*_____________) Direct ESCON Adapter (Part 1 of __) S/390 Information; Customer name:____________________________ 181
21. (*_____________) Direct ESCON Adapter (Part 2 of 2) S/390 Information; Customer name:______________________________ 182
22. Differences between SCSI-1 and SCSI-2 195
23. Differences between SCSI-2 and SCSI-3 196
24. SCSI Commands for Sequential-Access Devices 197
25. SCSI Commands for Medium Changer Devices 198

Preface

This redbook, *Data Sharing: Implementing InfoSpeed Copy-Sharing Solutions Among Heterogeneous Systems*, is written for people marketing or implementing data copy-sharing solutions, such as IBM System Engineers, IBM Storage Business Partners, as well as storage administrators, system programmers, system administrators, and technical support and operations managers and staff.

This redbook will help you install, tailor and configure the new IBM InfoSpeed 9398 Data Gateway, Models DG1 and MX1, with InfoSpeed exPress Data Mover, Version 1 Release 1, 5697-D61 and 5648-PDM.

This redbook gives a broad understanding of data sharing and detailed information on the new IBM data Copy-Sharing solution for heterogeneous systems.

This redbook will help you design and create data Copy-Sharing solutions between OS/390 and UNIX or Windows NT environments. The redbook provides detailed technical guidance for implementing the InfoSpeed solution on heterogeneous servers. It recommends different ways you can implement InfoSpeed and gives examples of how to copy data between the different environments.

The Team That Wrote This Redbook

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

Kjell E Nystrom is a Cross Storage Solutions Specialist at the International Technical Support Organization, San Jose Center. He writes extensively and teaches IBM classes worldwide on all areas of Storage Solutions. Before joining the ITSO in 1997, he worked for IBM Storage Systems Nordic Division, Sweden, as Storage Software Products Manager/Storage Marketing Support Specialist. Kjell has over 30 years of experience in S/390 and Storage Systems. His areas of expertise include all major hardware and software products for the S/390 platform, Storage Systems, data disaster recovery, and data sharing. Kjell can be reached at *kjellen@us.ibm.com*

Cyril Anam is a Storage Administrator with IBM Global Services in the Securities Industry Services (SIS) in Canada. He has two years of experience in the storage field and over ten years experience in MVS applications. Cyril holds a degree in commerce and economics from the University of Toronto. His areas of expertise include hardware and software products for the S/390 platform and MVS applications for SIS. Cyril can be reached at *canam@torvm3.vnet.ibm.com*

Marc Chapel, Ph.D., is an Open Systems Storage marketing support specialist in EMEA, based in Paris. He has worked for four years with ADSM, tape libraries, and disk subsystems in mainframe and open environments, and has implemented earlier data-sharing solutions in Europe. He has written extensively on ADSM and partitioning and sharing of tape libraries. Marc can be reached at *f065377@fr.ibm.com*

Keijo Ekman is a Storage Solution Specialist in the High End Sales Organization in Finland. He has over 30 years of experience in S/390 and Storage Systems.

His areas of expertise include all storage hardware and software products for the S/390 platform. He has participated in several residencies on storage hardware and software. Keijo can be reached at *kekman@fi.ibm.com*

Marty Gardner is a Staff Systems Consultant at Computer Network Technology, Inc., located at Minneapolis, MN, USA. He has over 35 years of experience in data communications and data processing in the mainframe and open systems environment. His areas of expertise include MVS and OS/390 systems and application processing as well as open systems OS, and application processing. Marty can be reached at *marty_gardner@cnt.com*

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

Thomas Connolly
IBM SSD, San Jose

Scott Drummond
IBM SSD, San Jose

Cindy Grossman
IBM SSD, Tucson

Jason Luurs
IBM SSD, Tucson

Thuy Nguyen
IBM SSD, San Jose

Valerie Poggi
IBM SSD, San Jose

Diane Williams
IBM ATS, San Jose

Bill Worthington
IBM ATS, San Jose

Harry Young
IBM SSD, Atlanta

Comments Welcome

Your comments are important to us!

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

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 231 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Chapter 1. Introduction to Data Sharing

As a broad generalization, access to data is provided by two different structures:

- Network-based data sharing

  Various structures and methods of operation can be used, typically characterized by the use of a communication program associated with the operating system and a monitor that provides application and/or user access to data.

- Storage subsystem-based data sharing

  When a storage subsystem is used for data sharing, there is a close relationship between the way the data is managed and the storage subsystem. Several interface programs are available to make it easier for the user and the application to access the data.

This book covers only storage subsystem-based sharing. For more information on network-based data sharing, please refer to *Data Sharing: Using the OS/390 Network File System*, SG24-5262.

With today′s increasingly sophisticated applications, such as business intelligence and e-business, information must be accessible to anyone who needs it. IBM′s InfoSpeed data-sharing solution uses an optimal, one-step, data-copy sharing approach to meet today′s challenges of open connectivity, flexibility, and performance.

This chapter gives an overview of data sharing and the three types of data sharing. The term *data sharing* describes the access of common data for processing by multiple computer platforms or processors. Different platforms store and access data differently. For example, disk storage for OS/390, UNIX, and Windows NT have different methods for data encoding, file structures, disk subsystem attachment mechanisms, and other data management protocols. Given the complexity of these differences, it is important to determine whether data is shared between platforms that are similar or different. The term *homogeneous data sharing* indicates that all platforms are the same; *heterogeneous data sharing* indicates at that least two platforms are different.

Figure 1 on page 2 shows the concept and types of data sharing.

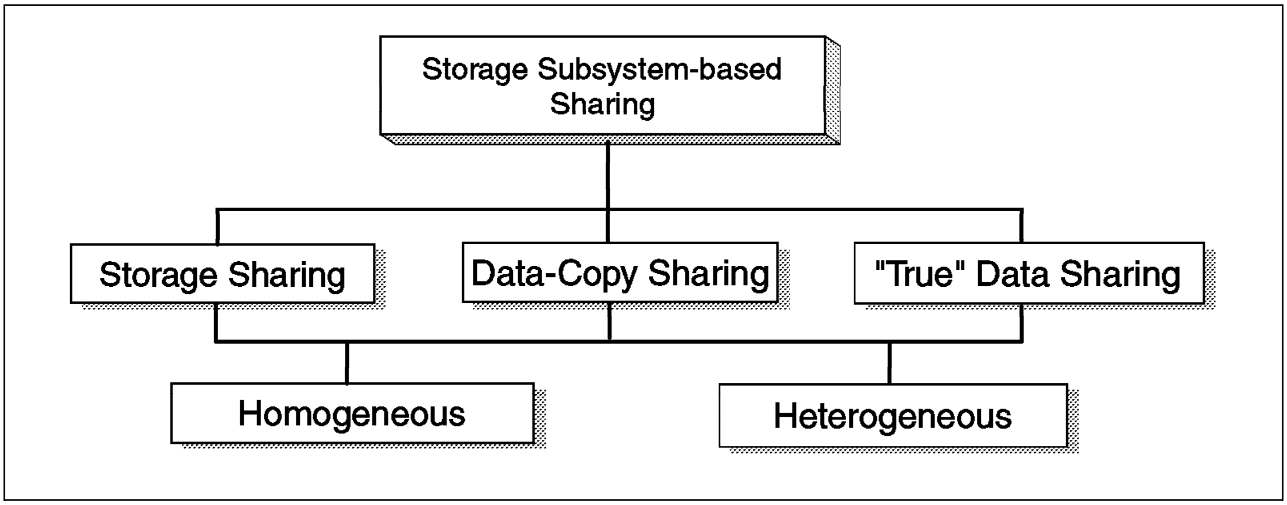


Figure 1. Data Sharing Concept and Types

1.1 Types of Data Sharing

The three types of data sharing entail different degrees of complexity and data accessibility among homogeneous or heterogeneous computer platforms. The three types of data sharing are:

- Storage sharing
- Data-copy sharing
- True data sharing

1.1.1 Storage Sharing

With storage sharing (Figure 2 on page 3), two or more homogeneous or heterogeneous platforms share a single storage subsystem whose capacity has been physically partitioned so that each attached platform can access only the units allocated to it. Multiple platforms can own the same partition, but this is possible only with homogeneous platforms such as in a S/390 Parallel Sysplex configuration.

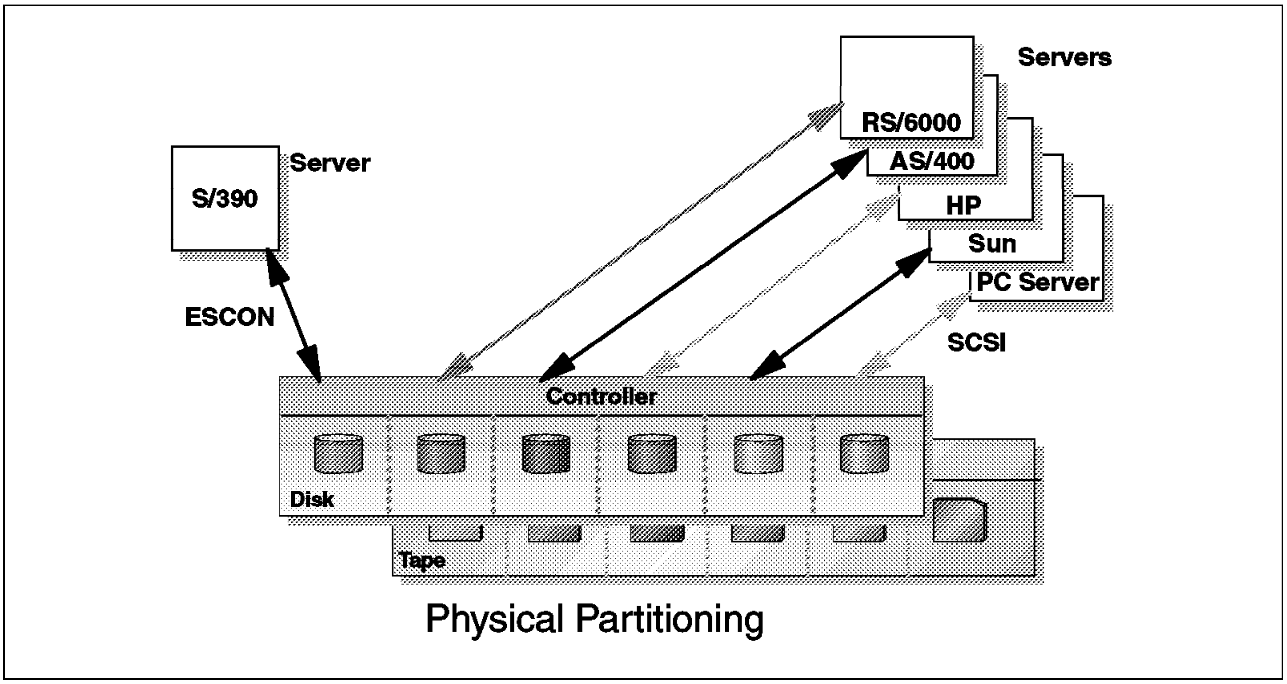


Figure 2. Example Showing How Storage Sharing Works

Different platform types use different attachment mechanisms for storage subsystems. For example, S/390 uses Enterprise Storage Connection (ESCON) channels, while UNIX systems or PC-based Windows NT servers typically use small computer storage interface (SCSI) attachments. See Appendix C, “Small Computer Systems Interface (SCSI) Tutorial” on page 187 for more information on SCSI attachments. To implement storage sharing, the storage subsystem must support each platform′s attachment mechanism, or a functional component must be placed between the processor and the storage subsystem that translates the commands and protocols used by the attachment mechanism.

1.1.2 Data-Copy Sharing

One method of data sharing among different platforms is to send a copy of the data from one platform to the other. This is known as *data-copy sharing*.

Data-copy sharing allows different platforms to access the same data. It requires all the copies to be made before the data is updated. The speed of the copy operation is important because data availability depends directly on how quickly the data can be copied.

There are two approaches to data-copy sharing between heterogeneous platforms: bulk file transfer and one-step data transfer. The bulk file transfer approach is typically a three-step serial process (see Figure 3 on page 4):

1. Extract the data to a flat file.
2. Transfer the flat file to another system.
3. Load the flat file into a database.

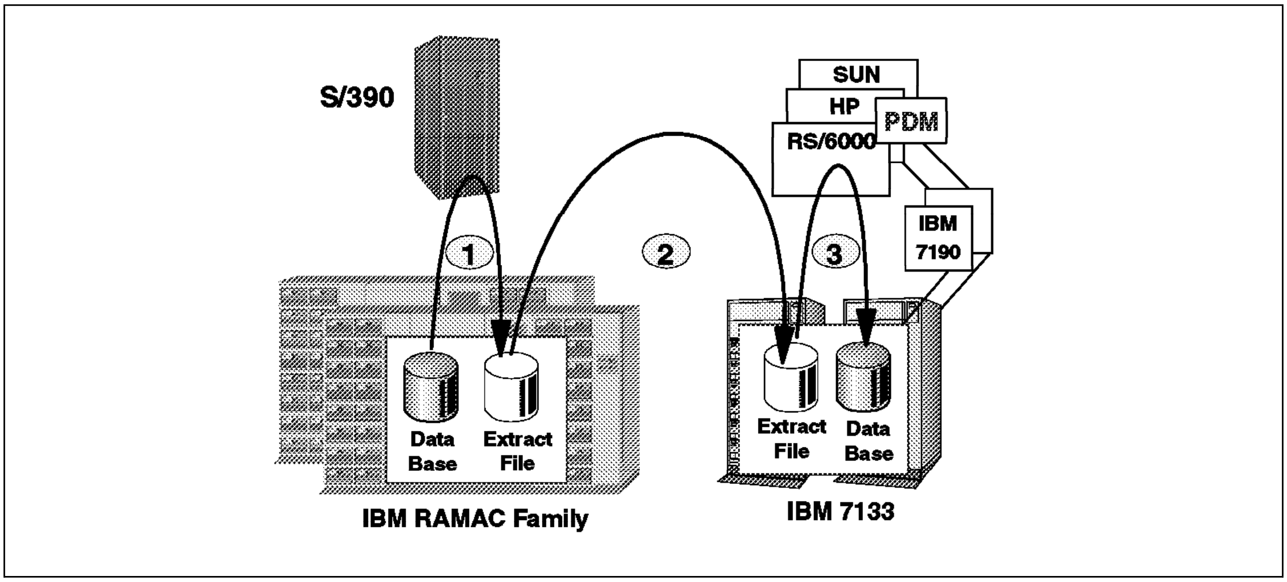


Figure 3. Data-Copy Sharing Using a Three-Step Process

The one-step data transfer approach involves a direct application-to-application data transfer (see Figure 4), which is significantly faster and cheaper than the bulk file transfer as the extract and load process runs in parallel and requires no intermediate flat files.

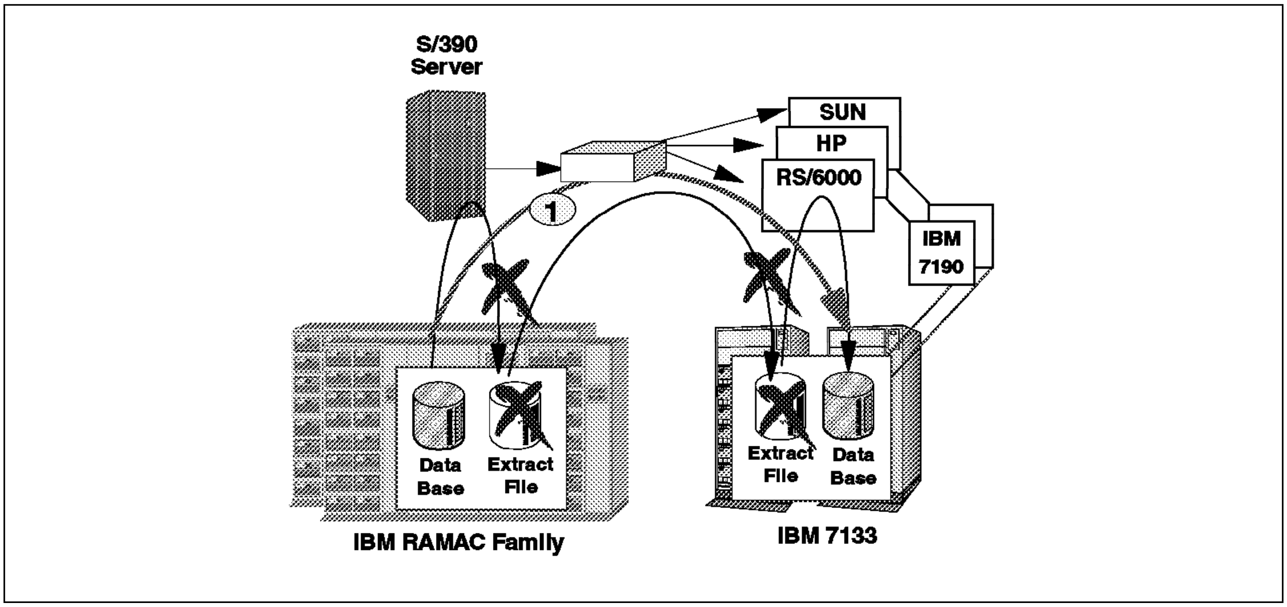


Figure 4. Data-Copy Sharing Using a One-Step Process

1.1.3 True Data Sharing

In true data sharing (Figure 5 on page 5), only one copy of the data is accessed by multiple computer platforms, whether homogeneous or heterogeneous. Because there is only one copy of the data, there is never a need to replicate the data for use elsewhere. Every platform attached has read and write access to the single copy of the data. True data sharing reduces maintenance and storage costs and eliminates any data synchronization problems.

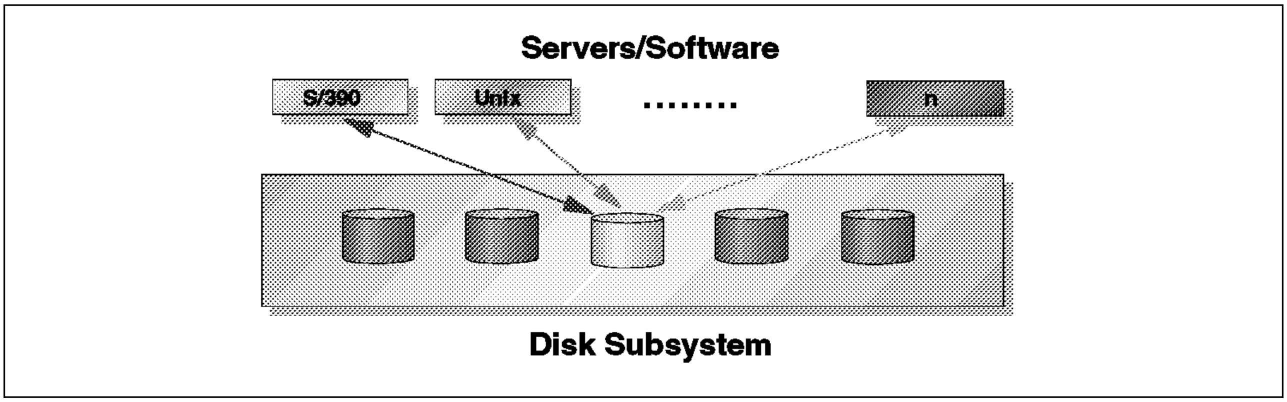


Figure 5. True Data Sharing

In today′s computing environment, true data sharing exists only on homogeneous platforms. For example, in an S/390 Parallel Sysplex configuration, all processors in the sysplex run on similar platforms and have read and write access to data on shared disk storage. Applications running on separate processors can simultaneously read the same copy of the data, but write access requires the data to be locked by a single application in order to preserve data integrity. The process of locking data is managed by hardware and software independent of the disk storage subsystem.

Another example of true data sharing is an RS/6000 cluster configuration with shared-disk architecture. Clustering technology allows a group of independent nodes to work together as a single system, and in a shared-disk architecture, each node can be connected to a shared pool of disks and its own local disks. All of the nodes have concurrent read and write access to the data stored on the shared disks. As with the S/390 Parallel Sysplex, write access requires the data to be locked by the node requesting the write to preserve data integrity. This locking process is also managed by software independent of the disk storage subsystem.

Figure 6 shows two examples of true data sharing in homogeneous environments.

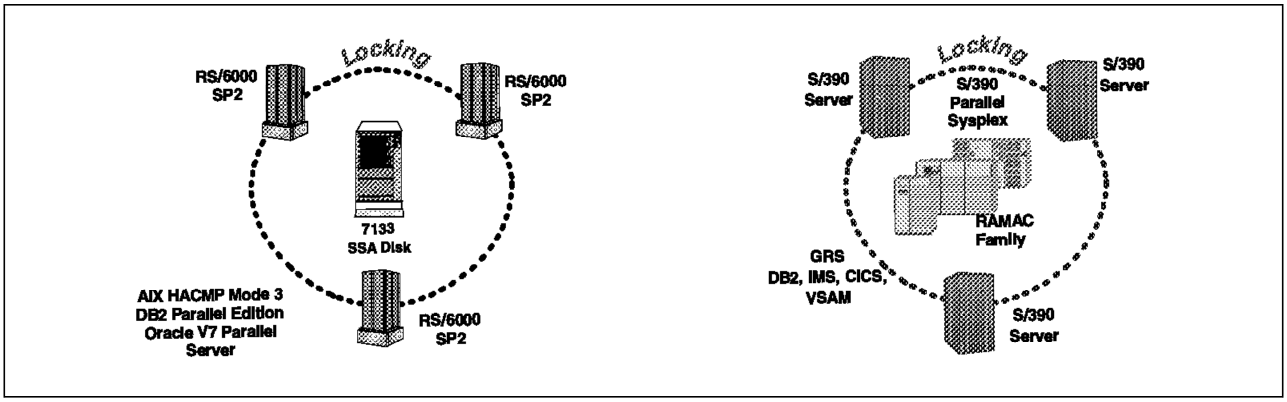


Figure 6. True Data Sharing in Homogeneous Environments

1.2 Business Values of the IBM InfoSpeed Solution

IBM InfoSpeed offers a high-speed piping solution that can directly transfer data from an OS/390 server database extraction program to a UNIX or Windows NT server database loader program. The IBM InfoSpeed solution (see Figure 7) is a one-step process that does not require the intermediate steps of flat file creation and movement. IBM exPress Data Mover (PDM) software and IBM InfoSpeed Data Gateway hardware make up the IBM InfoSpeed data-copy sharing solution.

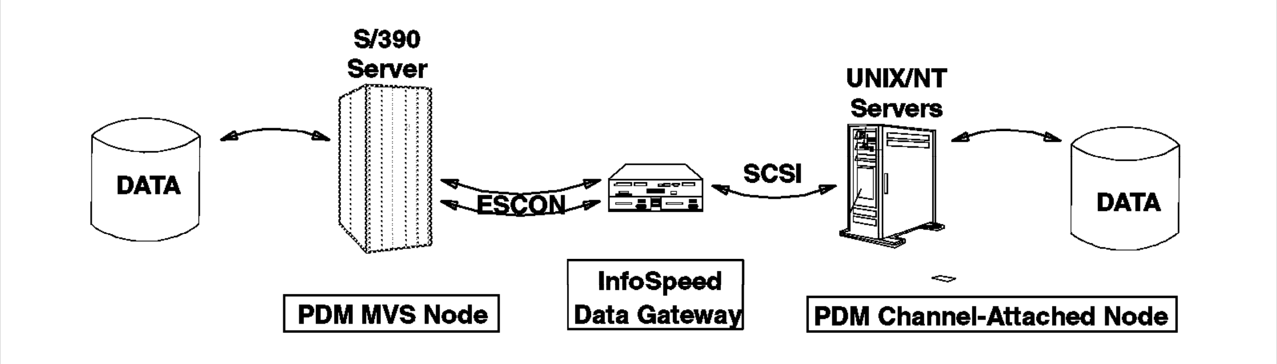


Figure 7. IBM InfoSpeed Solution

IBM InfoSpeed enables fast (up to 28 megabytes per second or 28MB/s), universal data sharing among S/390 and open-system servers through the use of dedicated channels and software that can transmit data between S/390, UNIX, and Windows NT platforms. Translation between ASCII and EBCDIC can be requested by the user. Fast data transfer is necessary for the best performance of enterprise applications such as data mining, business intelligence, and network computing. In most cases, the benefits can be achieved without changes to application programs.

Figure 8 on page 7 shows the connection options available for InfoSpeed at announcement time.

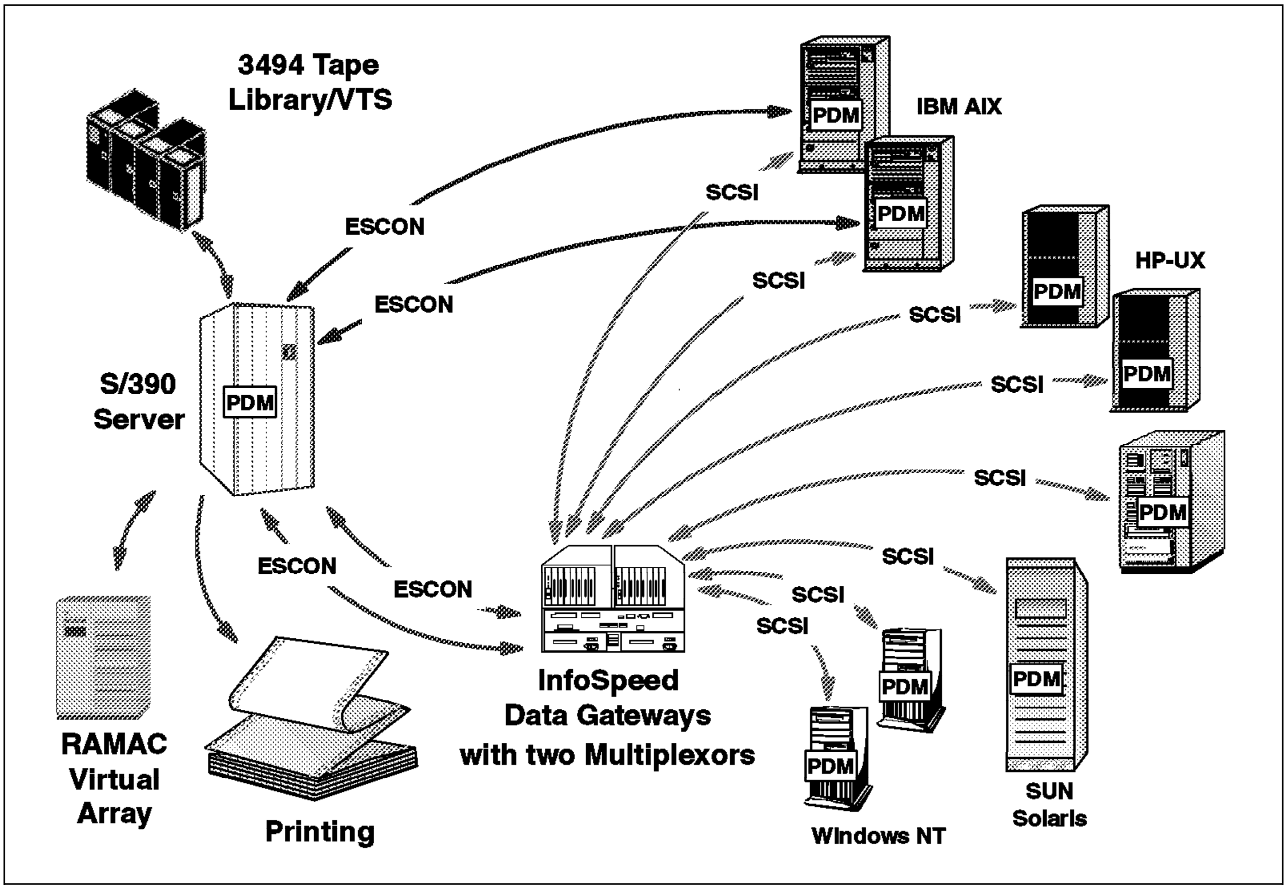


Figure 8. InfoSpeed Connection Options

Table 1 lists some of the information technology (IT) and business values of IBM InfoSpeed that to discussed in following sections.

Table 1. Information Technology Values and Business Values of IBM InfoSpeed

IT Values	Business Values
• Dedicated channel eliminates network bottlenecks • Eliminate TCP/IP processing and associated overhead costs and possible security exposures • High speed application-to-application data piping • Extend channel distances through use of ESCON channels • Lower disk capacity costs • Storage and server independent • VTS (virtual tape server) and tape management for Open Systems	• On-demand file transfer using dedicated channels • Provides more current data for enterprise applications such as data mining, business intelligence, and network computing to achieve competitive advantage • Universal data access between S/390 and open system servers • Data Gateway supports UNIX and NT servers • Enterprise-wide business continuance service • High-volume printing for Open Systems

1.2.1 InfoSpeed and Data Currency

The replicated data created by InfoSpeed is a separate copy from the original. It can quickly become different from the source data as changes are made to the two versions. Completing the replication process quickly and efficiently is essential to making sure the copied data is very close in content to the original, or as current as possible.

Another influence on performance in a data-copy sharing solution is the number of data copy steps required. Many of today′s solutions that use file transfer protocol (FTP) require three steps:

- Extracting data to a flat file at the sending server
- Copying extracted data to another flat file on the receiving server
- Loading the copied flat file into database format at the receiving server

The IBM exPress Data Mover (PDM) software support, part of the InfoSpeed solution, includes subsystem interface (SSI) capability that can eliminate time-consuming additional steps.

Instead of creating one or two intermediate copies of data, the SSI can create the final copy in one step. The SSI enables InfoSpeed implementation with no changes to existing application programs in many cases.

In addition to the number of copies and the speed of doing copies, another factor that affects data currency on the receiving system is the availability of the data to be copied on the source server. Because that data is often critical to on-line transaction processing, it is frequently unavailable for copying.

When InfoSpeed is combined with the unique SnapShot storage solution on the IBM 9393 RAMAC Virtual Array Storage, the unavailability of the source data on the sending server can be minimized to just a few seconds.

1.2.2 InfoSpeed Information Sharing

IBM has long been known for providing total solutions to customer′s business problems, and IBM InfoSpeed is a key component in providing a complete solution. If a customer is interested in creating and moving a copy of an on-line S/390 database to an open systems server with a minimum of downtime for the entire process, IBM can offer a unique solution. The source database could be located in an IBM RAMAC Virtual Array with IBM SnapShot Copy capability. Using SnapShot Copy, the S/390 server could make a copy of the source database in as little as a few seconds. After the SnapShot Copy operation is complete, the source database could be returned to active use while IBM exPress Data Mover is used to move the Snap copy from the S/390 server to the open systems server and ultimately to a high-performance IBM 7133 subsystem accessible from multiple open system servers.

1.2.3 InfoSpeed and Storage Management

I/T is expected to manage the storage growth with common management procedures and polices including backup/restore, migrate/recall, archiving and business recovery services. With storage sharing, storage subsystem assets can be managed more capably, protecting and enhancing your investment in them. When a single type of storage can be used in support of more than one platform, it can be purchased in higher quantities with greater discounts. There are also savings from having only one type of storage to administer and manage. Savings will be realized when it can be managed and administered using a single set of tools.

If one platform can do most of the management of storage, there are benefits:

- A single centralized tape library can be used across all platforms for backup.
- Common, enterprise-wide business continuation process can be set up.

1.2.4 InfoSpeed and Data Mining and Data Warehousing

For applications such as data mining and data warehousing, servers using different operating systems typically share the same data by sending a copy to another server. In this way, different servers can access data at the particular point in time when the copy was created. Data copying can provide significant competitive advantages when data warehouses are able to access new data on demand and improve the currency of data.

New data access solutions can provide operational value when

- Data copying is done through high-speed, dedicated ESCON channels which can offload busy networks and reduce TCP/IP processing load.
- Transferring large volumes of data quickly gives users the advantage of the most current information for decision-making and data mining applications.

1.2.5 InfoSpeed and High-Volume Printing

In many installations the OS/390 system has very advanced printers equipped for high volume printing. For example IBM 3900 can have special hardware attached making it possible to print for several hours without operator intervention.

With InfoSpeed installed, UNIX and Windows/NT systems can easily copy files to the OS/390 system to be printed using the equipment described above.

1.2.6 InfoSpeed and Use of Tape Library

Today 3494 Magstar Tape Library attaches to S/390 and multiple open servers including AS/400, RS/6000, and selected HP and SUN servers through SCSI channels. Magstar Virtual Tape Server (VTS) supports today attachment to S/390 servers with ESCON channels. VTS supports automated tape-cartridge handling and logical volume mounts, writes, demounts, and reads in the VTS disk buffer. This Seascape architecture solution with outboard storage server can be extended into the data-copy sharing solution in the future.

VTS is an excellent solution to store sequential data from open system servers. The UNIX/NT servers can, by using the InfoSpeed solution, write data to or read data from a VTS attached to the S/390 server. This is a high speed, automated data-copy sharing solution for VTS users.

1.2.7 Tape Management System for UNIX/NT

The use of InfoSpeed and IBM 3494 Magstar Tape Library or Virtual Tape Server with associated tape management system in S/390 environment offers tape management services for UNIX/NT environments when their data is stored in this library.

1.2.8 InfoSpeed and ADSM

The IBM ADSTAR Distributed Storage Manager (ADSM) software offerings provide availability management (backup/restore and disaster recovery) and space management to distributed UNIX and Windows NT servers through network connections to OS/390 platforms. An ADSM S/390 server can manage a network of ADSM clients by providing centralized management of a hierarchy of distributed UNIX and NT platforms including those supported by InfoSpeed

As part of the IBM InfoSpeed announcement, IBM also released a Statement of Direction that ADSM will be enhanced to exploit the new features and functions of IBM InfoSpeed. Many customers already use ADSM to provide backup/recovery, hierarchical storage management, and business continuity services for files on a wide variety of supported open system servers. ADSM currently uses telecommunications links to move data between the source servers and the backup servers. Many customers have expressed a desire for more efficient connectivity between source and backup servers, which IBM plans to provide using IBM exPress Data Mover software and InfoSpeed hardware. Not only will such connectivity offer high performance, but it will also off-load existing telecommunication links and eliminate the TCP/IP protocol overheads associated with such links.

1.2.9 InfoSpeed and Backup Solutions

Improved data backup and recovery is one of the benefits of data-copy sharing. UNIX/NT servers do not enjoy the same level of data management or sophisticated hardware found on most S/390 servers. Also, depending on the data backup procedures and techniques available on several UNIX/NT servers makes many customers uncomfortable. The data on those servers is rarely backed up on the high-availability, high-performance tape subsystems generally connected to S/390 servers. However, data-copy sharing techniques can be used to copy critical data from UNIX/NT servers to the S/390 server. There, data is backed up using rigorous, proven techniques and procedures typical of that environment.

High performance channel connections can also improve business recovery with faster data restoration.

1.2.10 Create an Information Network

Many customers would like to improve connectivity between S/390 servers and open system servers. A single IBM InfoSpeed Data Gateway can be used to provide an information network to move files from an open system server to an S/390 server, and even through the S/390 server to another open system server. This new connectivity option offers exciting possibilities for extending and enhancing data management capabilities, and to do so without requiring any application program changes.

1.2.11 InfoSpeed and Data Archive Solutions

A tape library in an S/390 environment offers a good solution for archiving data from the UNIX/NT environment for a longer period. VTS offers an even better solution than does the native Magstar library in terms of tape unit availability and media capacity usage. In an SMS environment, selecting the right management class provides a long retention period for data, which can be seen as a minimum archive solution. For more advanced archiving, there is a need for a catalog. The ADSM archiving function offers excellent catalog services.

1.2.12 InfoSpeed Benefit Summary

It is easy to summarize the many benefits of the IBM InfoSpeed solution. High-performance cross-platform data sharing is probably the most significant benefit, followed closely by the flexible and efficient new data piping capability, which requires no application program changes and is easy to automate. Many customers also consider server and storage vendor independence to be as significant a benefit as telecommunications network off-load and elimination of associated TCP/IP overheads. With these significant improvements, many customers should be anxious to install IBM exPress Data Mover software and IBM InfoSpeed Data Gateway hardware and begin enjoying these benefits immediately.

Chapter 2. IBM InfoSpeed Technical Overview

Figure 9 shows the components of an IBM InfoSpeed solution.

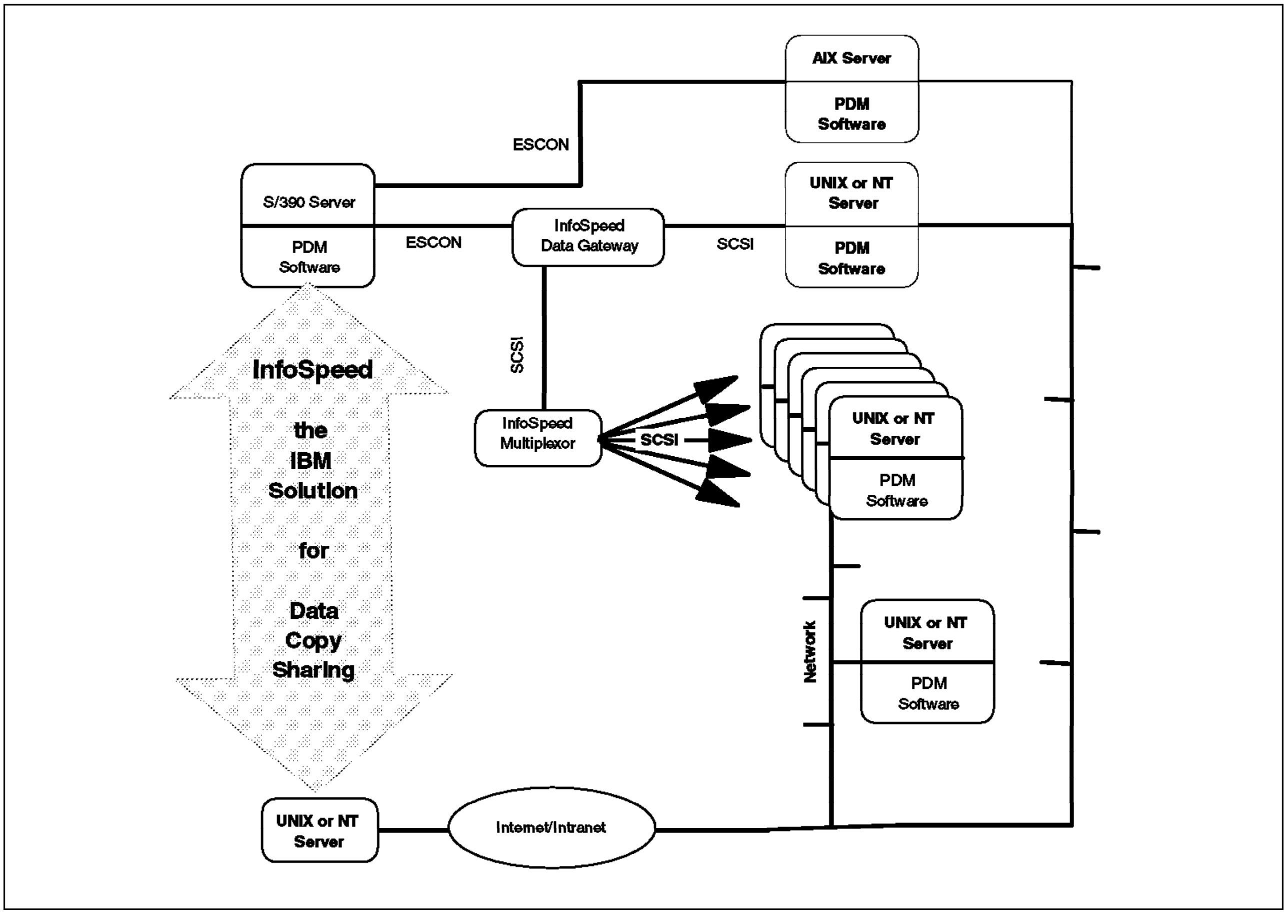


Figure 9. Components of an IBM InfoSpeed Solution

IBM InfoSpeed is a storage-based solution for data sharing, transferring data using ESCON channels and SCSI buses instead of networks. IBM InfoSpeed supports data-copy sharing that is, the replication of data from one server to another, without the use of intermediate flat files.

The IBM InfoSpeed product consists of three basic components: The IBM InfoSpeed Data Gateway (IBM 9398-DG1), the IBM InfoSpeed Multiplexor (IBM 9398-MX1) and the IBM exPress Data Mover software: IBM exPress Data Mover Version 1 for the OS/390 platform, 5697-D61 and IBM exPress Data Mover Version 1 for for the UNIX and NT platforms 5648-PDM. The IBM InfoSpeed Data Gateway and the IBM InfoSpeed Multiplexor are not mandatory; their use depends on the configuration installed. If an ESCON channel has already been installed between the S/390 and the UNIX machine (currently AIX only), only IBM exPress Data Mover software is required.

IBM exPress Data Mover software is mandatory on the S/390 side as well as on the UNIX and Windows NT side. UNIX and Windows NT servers that do not have IBM exPress Data Mover installed can also access data on the S/390 server, through their TCP/IP file transfer protocol (FTP), as described in Chapter 4, “IBM

InfoSpeed: Concepts and Principles” on page 25 and Chapter 10, “Using MVS PDM Clients” on page 105.

In this chapter, we describe the three basic components of IBM InfoSpeed. IBM exPress Data Mover concepts and principles are also discussed in Chapter 4, “IBM InfoSpeed: Concepts and Principles” on page 25.

2.1 IBM InfoSpeed Data Gateway

The IBM InfoSpeed Data Gateway Model DG1 is a SCSI-ESCON converter. Figure 10 shows the front face of the IBM InfoSpeed Data Gateway where all connections are located.

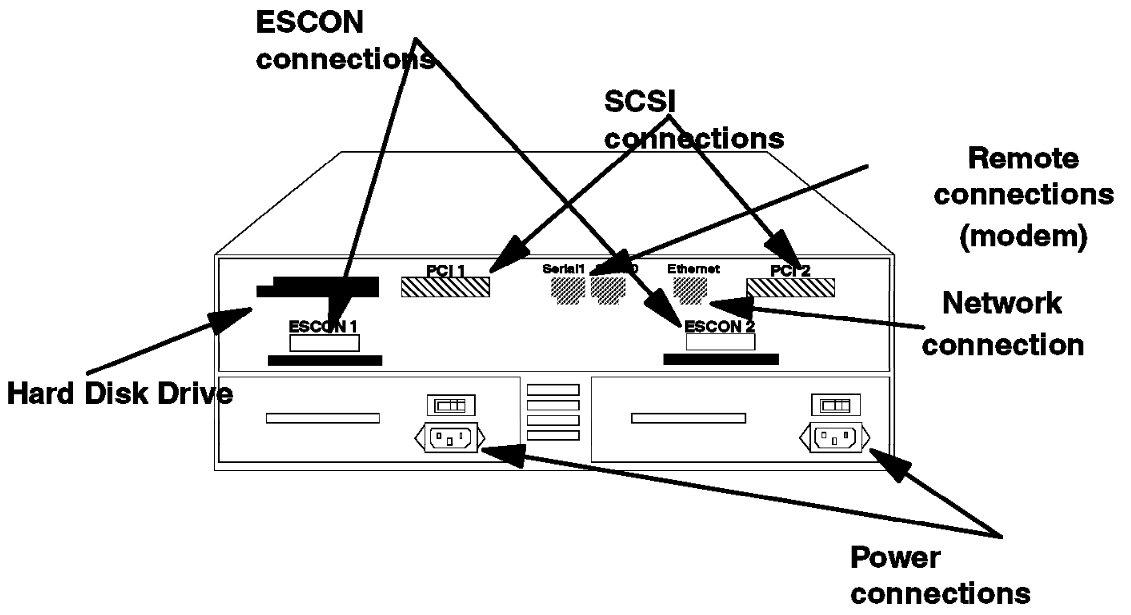


Figure 10. IBM InfoSpeed Data Gateway Model DG1

With at least one SCSI port and one ESCON port to transfer data, the DG1 allows dual connection to one S/390 server and one UNIX or Windows NT server. The IBM InfoSpeed Data Gateway can support multiple parallel data transfers in both directions for optimal performance. A second ESCON port and a second SCSI port can be added for higher throughput. Parallelism of multiple data transfers is possible through the use of multiple data paths and control paths on the ESCON and SCSI channels.

2.2 IBM InfoSpeed Multiplexor

Figure 11 shows the front face of the IBM InfoSpeed Multiplexor where all of the SCSI connections are located.

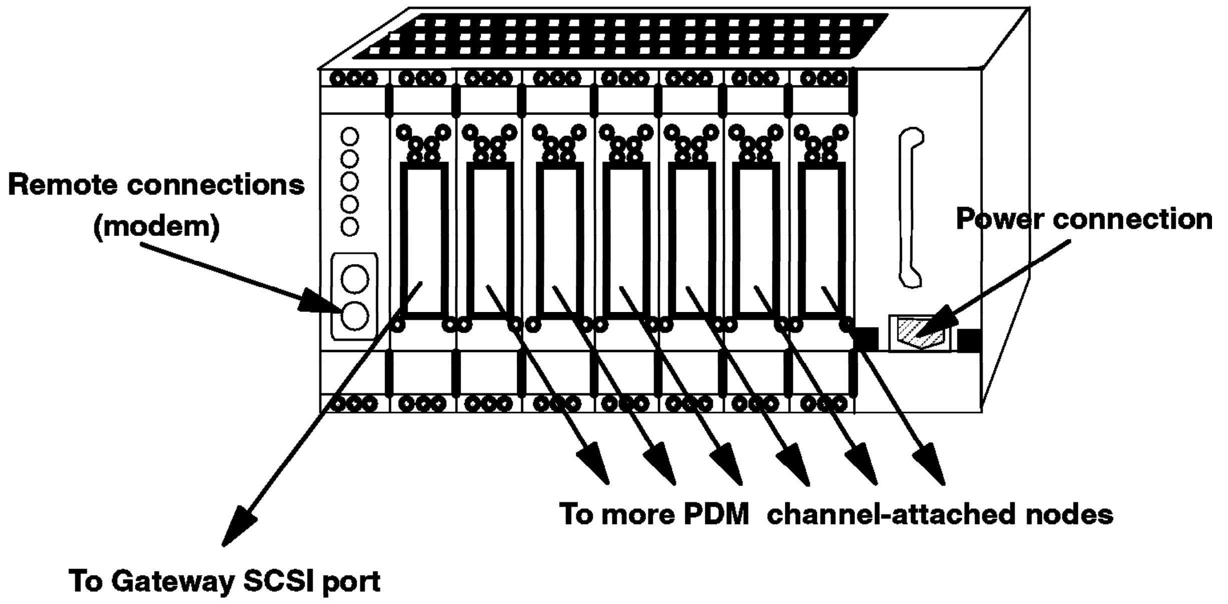


Figure 11. IBM InfoSpeed Multiplexor Model MX1

The IBM InfoSpeed Multiplexor Model MX1 switches several (two to six) SCSI port connections to a single SCSI port on the Model DG1. Connecting two IBM InfoSpeed Multiplexors to one IBM InfoSpeed Data Gateway enables up to 12 connections in the Data Gateway system, thus enhancing Model MX1 connectivity to UNIX and Windows NT open systems.

2.3 IBM exPress Data Mover

IBM exPress Data Mover is the key component of IBM InfoSpeed. It transmits data between S/390 and UNIX or Windows NT platforms. Hardware (such as the IBM InfoSpeed Data Gateway or IBM InfoSpeed Multiplexor) is required only when a customer′s UNIX or Windows NT server does not support ESCON attachment. In that case, IBM InfoSpeed Data Gateway Model DG1 can be used.

As shown on Figure 12 on page 16, the IBM exPress Data Mover (PDM) software has to be installed on both the S/390 and the UNIX channel-attached host.

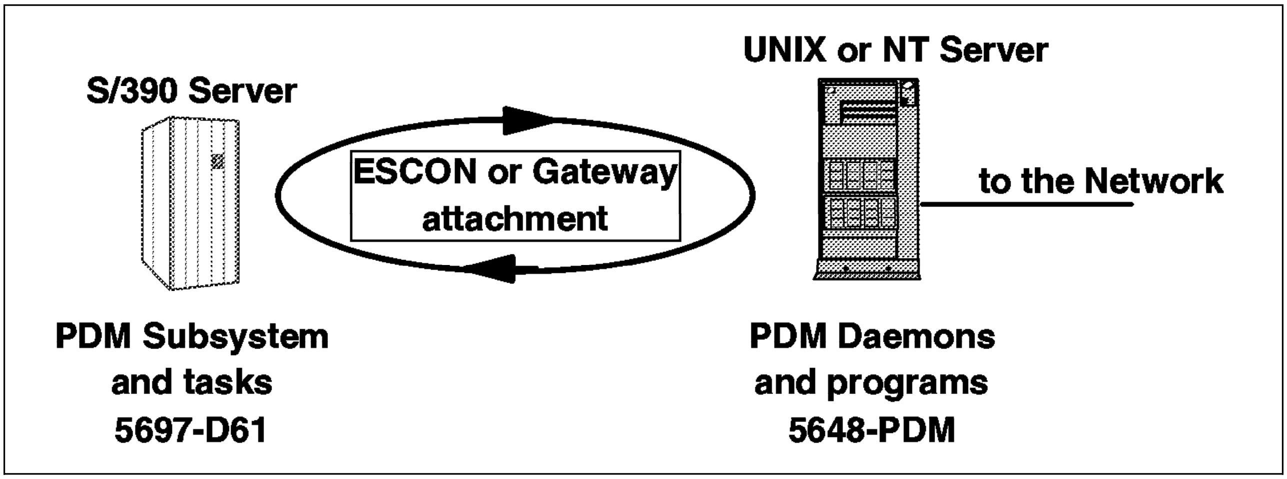

UNIX or NT Server
S/390 Server
ESCON or Gateway attachment
to the Network
PDM Subsystem and tasks 5697-D61
PDM Daemons and programs 5648-PDM

Figure 12. Configuration with the IBM exPress Data Mover

The PDM MVS subsystem and tasks are responsible for converting EBCDIC to ASCII data, for handling communications through ESCON channels, and for the interface with the OS/390 supervisor and end users.

The PDM UNIX or Windows NT software and programs are responsible for handling the communications through the ESCON channels or the SCSI buses, through both interfaces with UNIX or NT systems, and (optionally) through TCP/IP.

UNIX and NT machines that are not connected to an S/390 through an ESCON channel or an IBM InfoSpeed Data Gateway can still access S/390 data as long as the IBM exPress Data Mover software is installed. See Chapter 3, “IBM InfoSpeed: Configuration Alternatives” on page 19 for details.

UNIX and NT machines that do not have the IBM exPress Data Mover software installed can also access S/390 data by using the standard FTP of TCP/IP. See Chapter 3, “IBM InfoSpeed: Configuration Alternatives” on page 19 for details.

2.4 Hardware and Software Requirements

The installation of the IBM InfoSpeed requires the presence of the following hardware and software products.

2.4.1 S/390 Server

2.4.1.1 Hardware

System/390 is required for all Parallel, Enterprise, and Multiprise Servers.

2.4.1.2 Software

Required software includes the following:

- MVS/ESA SP Version 5 Release 2.2 (5655-068 or 5655-069), or later
- OS/390 R3 (5645-001), or later

Resource Access Control Facility (RACF) or other security product supporting third-party checking may be required depending on the service level installed for

IBM exPress Data Mover. For more information, see Appendix B, “OS/390 Security Processing in IBM exPress Data Mover” on page 185.

2.4.2 RS/6000 and SP

2.4.2.1 Hardware

Required hardware consists of:

- ESCON Control Unit Adapter, feature 2756 (if directly connected to OS/390 ESCON subchannel)
- SCSI Adapter (if connected to the IBM InfoSpeed Data Gateway)

2.4.2.2 Software

Required software is AIX 4.2. The IBM exPress Data Mover has been tested with the following materials: RS/6000 PS580/E30, AIX 4.2, and adapter IBM FC2412. Support for the adapter IBM FC2416 is expected to be available later.

2.4.3 Hewlett-Packard 9000-800

2.4.3.1 Hardware

The SCSI Adapter is required

2.4.3.2 Software

Required software is HP-UX Version 10.2, 10.20. The IBM exPress Data Mover has been tested with the following materials: HP/9000-D220/D270-D271, HP-UX 10.20, and adapters HP 28696A, HSC FWD A4107A-0D1, and EISA SE A2679A.

2.4.4 Sun Microsystems Ultra Enterprise

2.4.4.1 Hardware

The SCSI Adapter is required.

2.4.4.2 Software

Solaris Versions 2.4, 2.5, or 2.5.1 is required. As an example, the IBM exPress Data Mover has been tested with the following materials: Sun Sparc5, Solaris 2.5, and adapter Sun X1062A.

2.4.5 PC Servers

2.4.5.1 Hardware

The SCSI Adapter is required.

2.4.5.2 Software

Microsoft Windows NT Version 4.0 NT Version 4.0 is required. As an example, the IBM exPress Data Mover has been tested with the following materials: Dell Dimension, Windows NT 4.00.1381, and adapters Adaptec AHA-2944W and Adaptec AHA-2940U.

Chapter 3. IBM InfoSpeed: Configuration Alternatives

In this chapter, we describe the different configurations you can build using IBM InfoSpeed, with or without the hardware features (IBM InfoSpeed Data Gateway and IBM InfoSpeed Multiplexor), and with or without the software feature (IBM exPress Data Mover).

IBM InfoSpeed is a flexible solution. However the IBM exPress Data Mover software must be installed on the S/390 system and on the UNIX or Windows NT machines that are channel-attached.

3.1 Direct ESCON Channel Attachment

Figure 13 shows a typical basic IBM InfoSpeed configuration, with direct ESCON channel attachments.

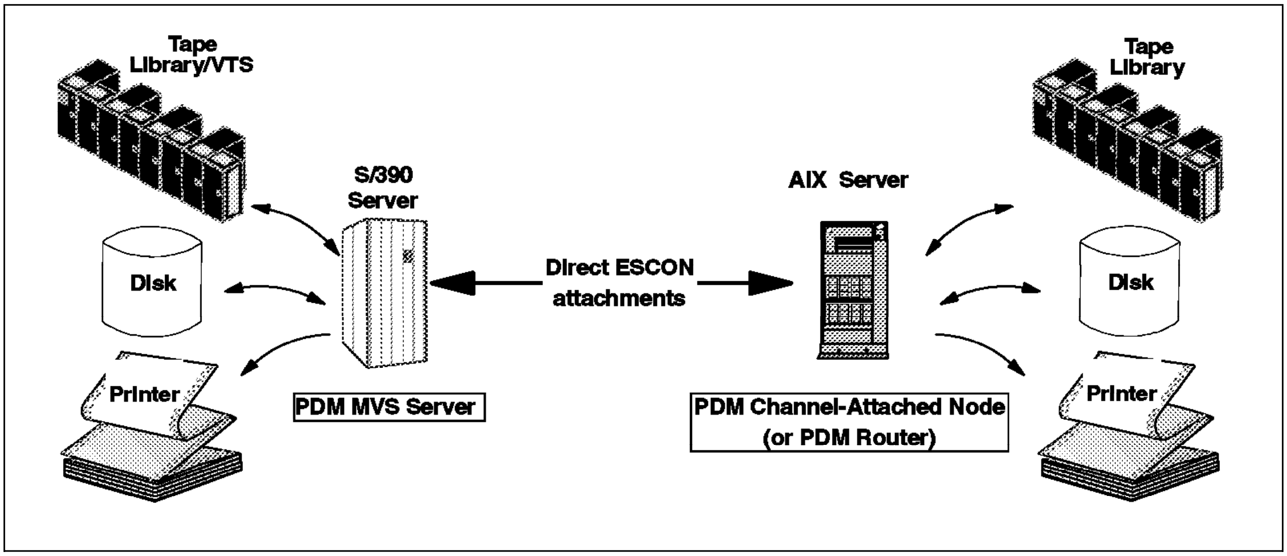


Figure 13. IBM InfoSpeed Basic Configuration

In the configuration in Figure 13, there is an ESCON channel attachment on both the S/390 and UNIX (currently AIX only) machines. The IBM exPress Data Mover software can use the direct ESCON attachment, and there is no need for an IBM InfoSpeed Data Gateway or an IBM InfoSpeed Multiplexor.

IBM exPress Data Mover software must be installed on both hosts. The S/390 host is called the *PDM MVS server*. The UNIX host is called the *PDM channel-attached node* or *PDM node router*.

This configuration provides peer-to-peer communication; it is possible, at the same time, to have other UNIX or Windows NT PDM channel-attached nodes connected to the same PDM MVS server, but through the IBM InfoSpeed Data Gateway. The ESCON channel attachment can also be shared by several OS/390 images if attached through an ESCON director. The ESCON adapter in the RS/6000 or SP, available at announcement of the IBM InfoSpeed, does not support the ESCON Multiple Image Facility (EMIF). The ESCON channel attachment cannot take advantage of the high performance (up to 28 MB/s) that you can achieve using the IBM InfoSpeed Data Gateway. The ESCON channel attachment is limited to no more than 13MB/s per attachment.

3.2 IBM InfoSpeed Data Gateway Attachment

Figure 14 shows an IBM InfoSpeed Data Gateway configuration with one IBM InfoSpeed Data Gateway installed.

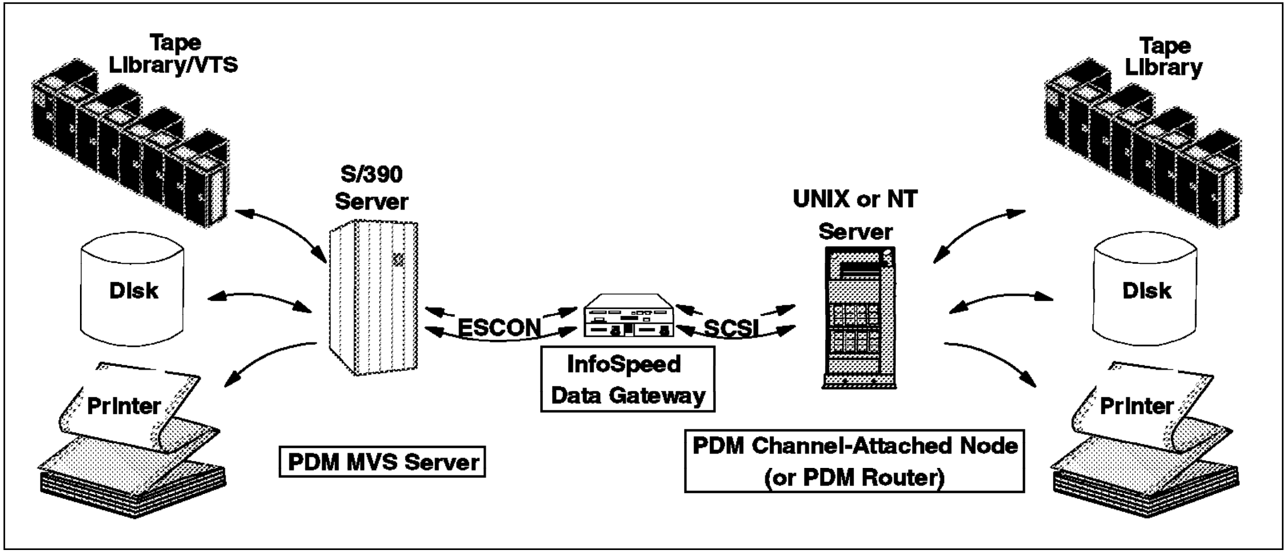


Figure 14. IBM InfoSpeed Configuration with One IBM InfoSpeed Data Gateway

In this configuration, the IBM InfoSpeed Data Gateway can be configured with either one or two ESCON channels and one or two SCSI buses. Each ESCON channel provides one-to-one communication with a PDM MVS server. Each SCSI bus provides one-to-one communication with a PDM channel-attached node. Many different arrangements can be configured as shown in Figure 15 on page 21.

The configuration shown in Figure 14 offers a performance of up to 28 MB/s. If more bandwidth is required multiple IBM InfoSpeed Data Gateways can be used between an OS/390 server and a single UNIX or Windows NT server.

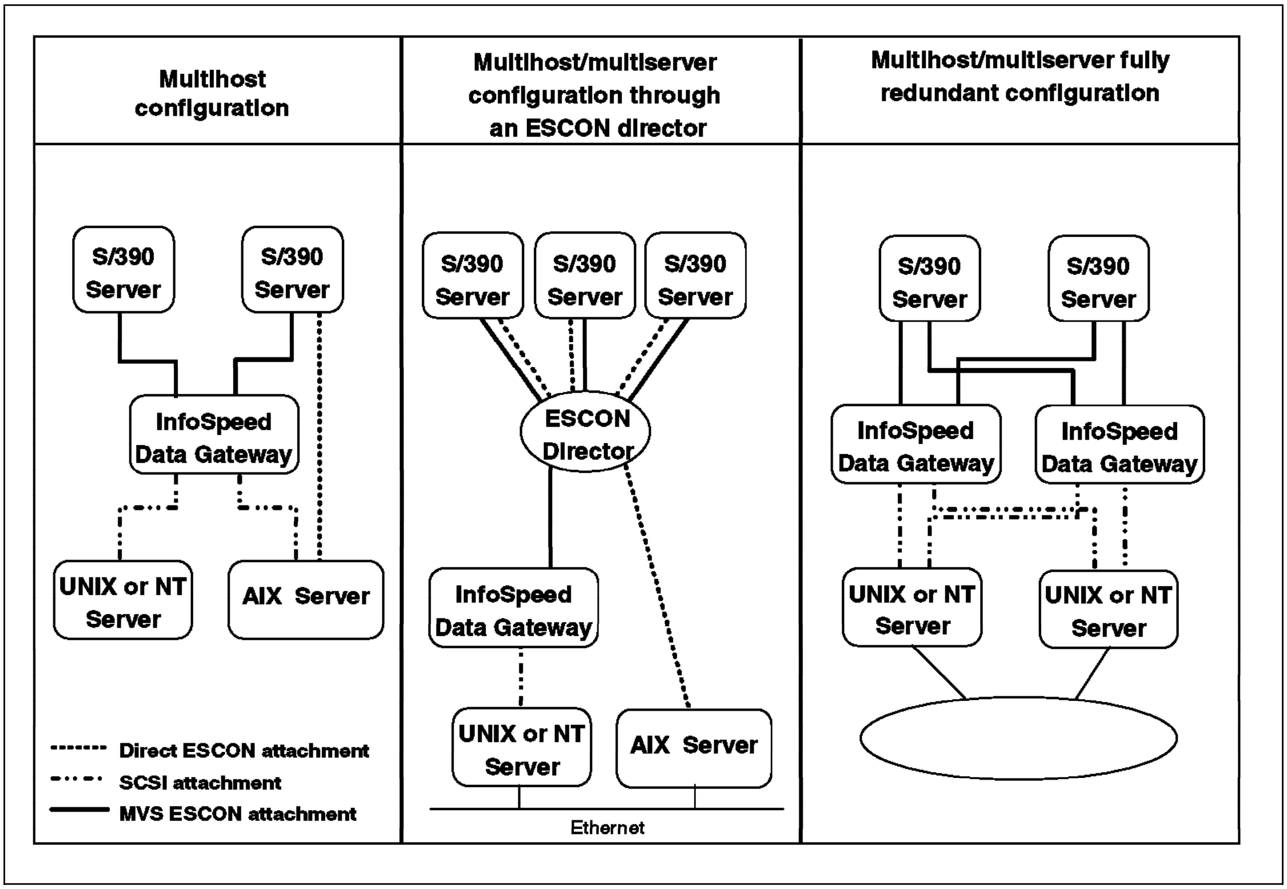


Figure 15. Different Configurations Using One or More IBM InfoSpeed Data Gateways.

The configurations in Figure 14 on page 20 and in Figure 15 take full advantage of parallel data transfers and high performance but continue to provide one-to-one communication. In all cases, the IBM exPress Data Mover software must be installed on all PDM MVS nodes and PDM channel-attached nodes.

The multihost configurations in Figure 15 show how a UNIX or NT channel-attached server can access two different 0S/390 servers through a unique gateway, by connecting the two ESCON connectors to two different OS/390 servers. A single OS/390 server can also use an ESCON attachment to the gateway in parallel with one or more direct ESCON attachment to one or more AIX servers.

The IBM exPress Data Mover supports EMIF, and the ESCON connections to the IBM InfoSpeed Data Gateway can be shared among multiple logical partitions, LPARs, in an S/390 server. An ESCON director can be considered as a single entry point to different OS/390 servers, attaching multiple OS/390 images running in multiple S/390 servers.

The redundant configuration can also be used to secure the communication paths between several OS/390 and UNIX or NT servers. If a direct connection to a server is not available, the IBM exPress Data Mover software will, use the network transparently as an alternate path, if so configured. ESCON directors can also be added to this configuration.

From Figure 15, you can speculate that it is also possible to establish data transfer paths from one UNIX or NT server to another. An intermediate OS/390 server has to be defined, but no intermediate file needs to be created. Such an example is shown in 12.2.3.8, "Copying Data Between UNIX or Windows NT Disks via MVS" on page 155.

The limitation to the number of OS/390 servers that can connected to the IBM exPress Data Mover is limited only by the capacity of the ESCON director itself, and by the number of ESCON channel addresses that can be defined on the OS/390 servers.

3.3 Adding the IBM InfoSpeed Multiplexor

Figure 16 shows a complete IBM InfoSpeed configuration, with one IBM InfoSpeed Data Gateway and two IBM InfoSpeed Multiplexors installed. This setup permits the connection of 12 PDM channel-attached nodes.

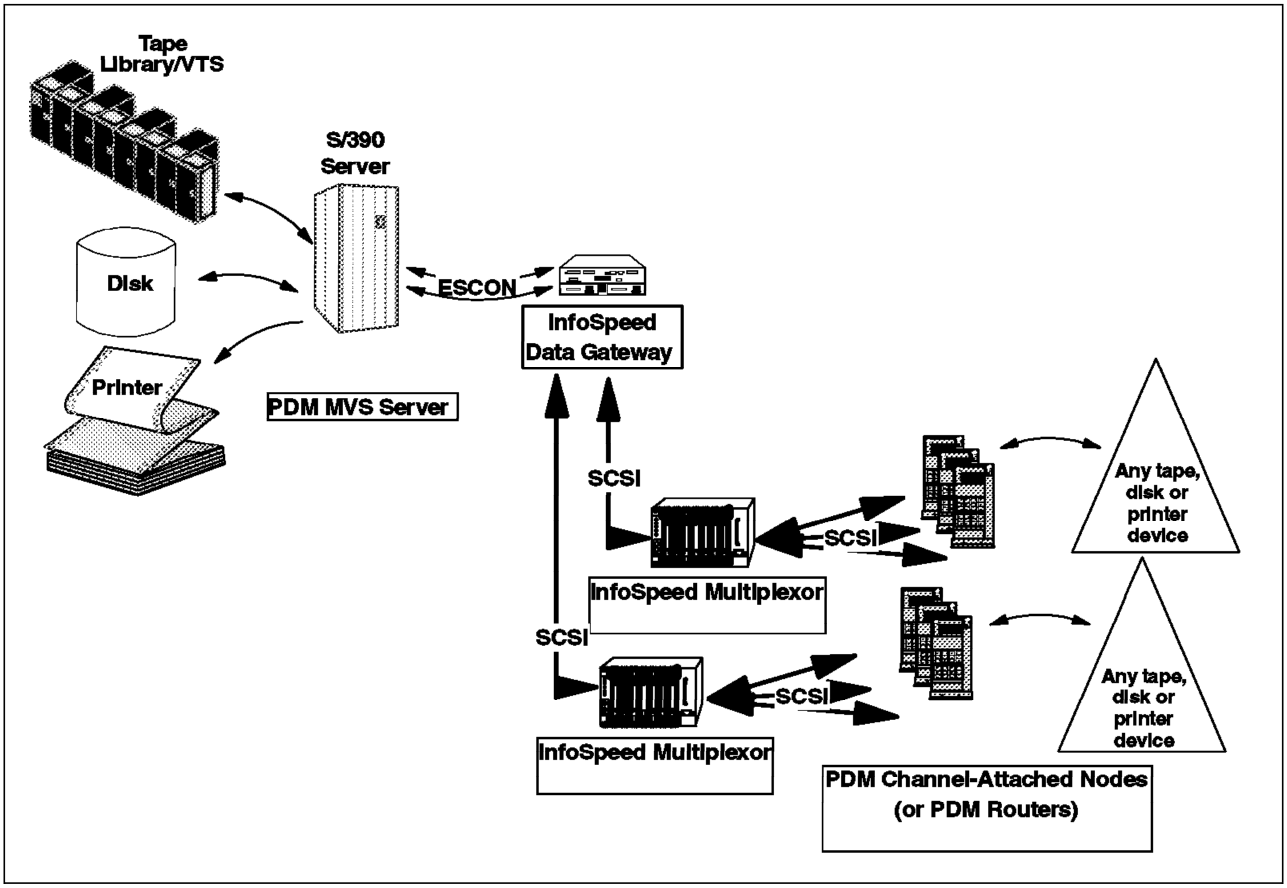
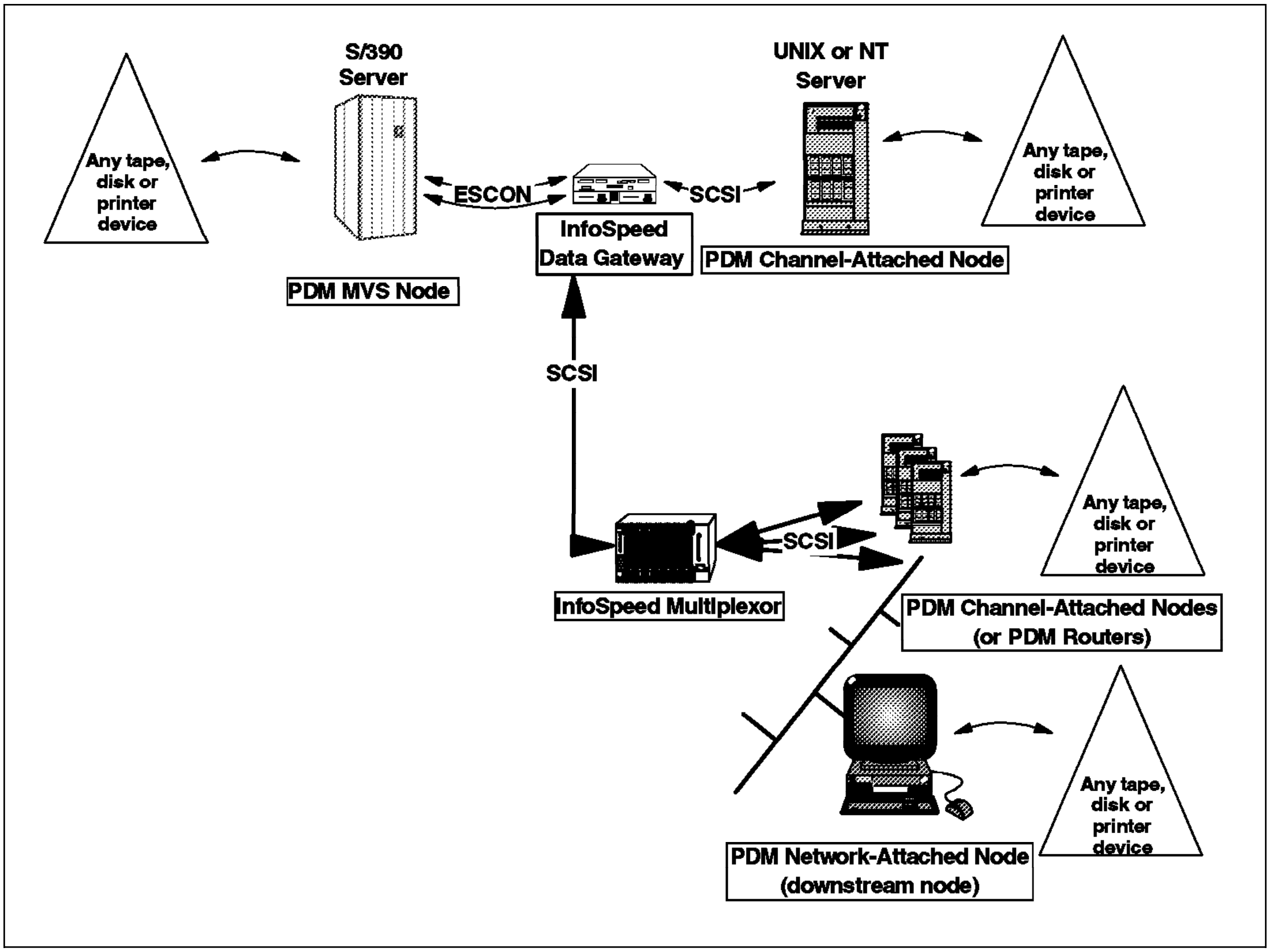


Figure 16. Full IBM InfoSpeed Configuration

In this configuration, the IBM InfoSpeed Data Gateway is connected through two SCSI buses to two IBM InfoSpeed Multiplexors, thus providing 12 ports to connect other PDM channel-attached nodes. Every PDM channel-attached node needs to have the IBM exPress Data Mover software installed. This configuration will offers a throughput of up to 26MB/s.

3.4 Adding PDM Network-Attached Nodes

Figure 17 shows a configuration where the IBM exPress Data Mover software has been installed on UNIX or Windows NT nodes, even though they are not channel-attached.



S/390 Server
UNIX or NT Server
Any tape, disk or printer device
ESCON
SCSI
InfoSpeed Data Gateway
PDM Channel-Attached Node
PDM MVS Node
SCSI
Any tape, disk or printer device
SCSI
InfoSpeed Multiplexor
PDM Channel-Attached Nodes (or PDM Routers)
Any tape, disk or printer device
PDM Network-Attached Node (downstream node)
Any tape, disk or printer device

Figure 17. Adding PDM Network-Attached Nodes to a Basic Configuration

In this example, only one IBM InfoSpeed Multiplexor is installed.

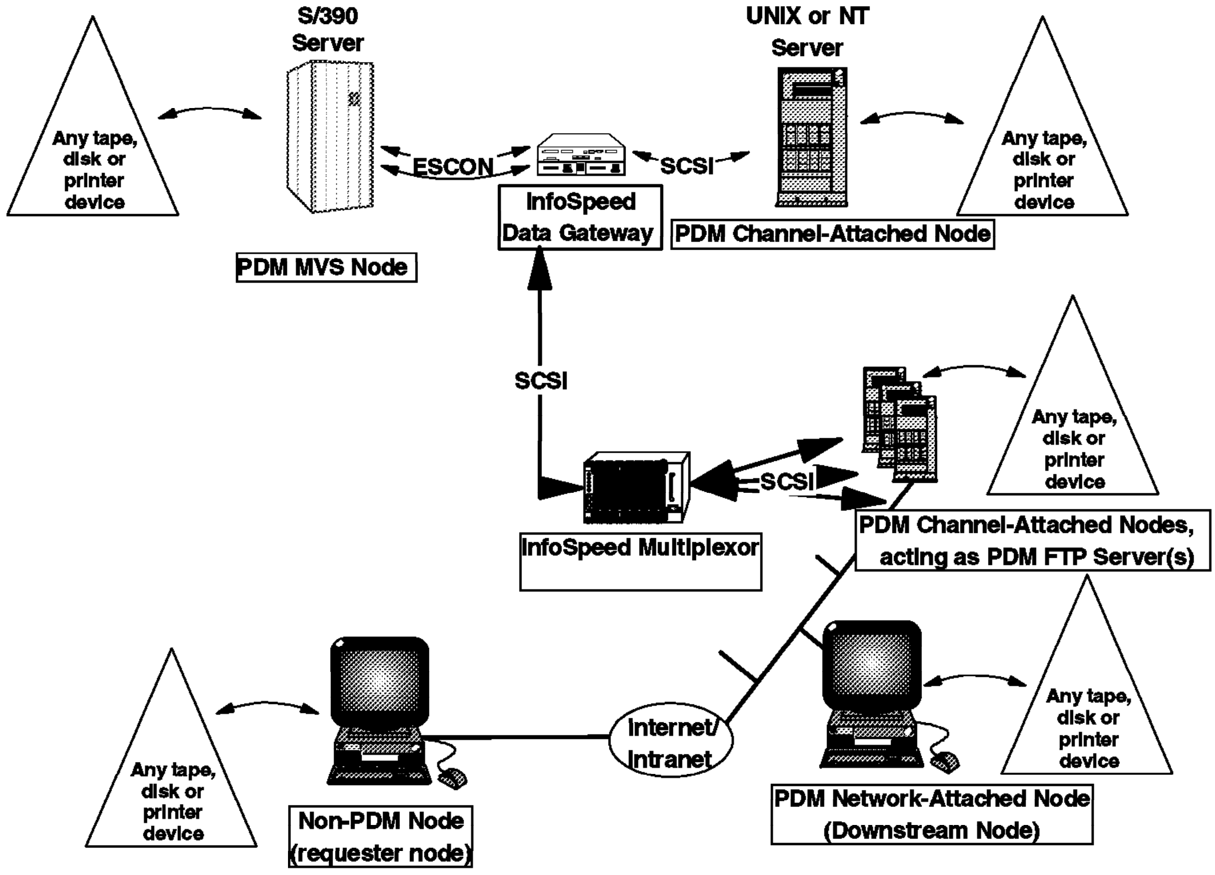
The PDM network-attached nodes can also access data on the PDM MVS server or be accessed by it through the PDM channel-attached node, which plays the role of a router.

The implementation and use of the PDM network-attached nodes are described in detail in Chapter 8, “Install and Configure PDM Network-Attached or Downstream Nodes” on page 97. The performance of the PDM network-attached nodes depends on the performance of the network. Performance details for different IBM InfoSpeed configurations are given in Chapter 11, “Performance Expectations” on page 135.

3.5 Adding Non-PDM Nodes

Some nodes, called non-PDM nodes, can also access the PDM MVS server or be accessed by it through the PDM channel-attached node (or PDM node router), without having the IBM exPress Data Mover software installed. These non-PDM nodes use the standard FTP protocol of TCP/IP, and a specific communication port.

Figure 18 shows a sophisticated configuration where nodes can communicate with the PDM MVS server, through the Internet, by using a PDM TCP/IP FTP server.


S/390
Server
UNIX or NT
Server
Any tape, disk or printer device
ESCON
SCSI
InfoSpeed
Data Gateway
PDM Channel-Attached Node
Any tape, disk or printer device
PDM MVS Node
SCSI
Any tape, disk or printer device
SCSI
InfoSpeed Multiplexor
PDM Channel-Attached Nodes,
acting as PDM FTP Server(s)
Any tape, disk or printer device
Internet/
Intranet
Any tape, disk or printer device
Non-PDM Node
(requester node)
PDM Network-Attached Node
(Downstream Node)

to a Base Configuration with PDM nodes in Place

Figure 18. Adding Non-PDM Nodes

The non-PDM node accesses the PDM MVS server through Internet or intranet, using a configured PDM FTP server. The FTP server acts as a PDM node transferring the communications from the requester node to the PDM MVS server through the PDM channel-attached node. The PDM channel-attached node then opens a session with the PDM MVS server.

The implementation and use of a non-PDM node are described in detail in Chapter 9, “Configure a PDM FTP Server for Non-PDM Nodes” on page 103.

Chapter 4. IBM InfoSpeed: Concepts and Principles

The IBM exPress Data Mover modules are installed on the mainframe and the appropriate UNIX or Windows NT servers.

The IBM InfoSpeed Data Gateway connects SCSI ports from the PDM UNIX or Windows NT software to ESCON mainframe channels. Servers that have an ESCON interface can attach directly to the mainframe.

IBM exPress Data Mover is OS/390-centric. When IBM exPress Data Mover Version 1.1 is used, all transactions are, in fact, initiated by OS/390, even if the request comes from a UNIX or Windows NT end user. The basic system for file transfers includes the PDM MVS subsystem on the mainframe and the PDM UNIX or Windows NT software. The PDM File Transfer Client is added to each end to allow file transfers to be initiated by either OS/390 or UNIX or Windows NT. The PDM File Transfer Client offers an FTP-like standard interface, with additional options such as checking on job transfer status.

This principle is used to copy extracted data in sequential file format, directly between the platform′s storage media (tape or disk). Multiple transfers can be initiated in parallel to achieve maximum data throughput.

4.1 The PDM MVS Subsystem

The PDM MVS subsystem consists of several components. Each component has a specific role:

- The PDM MVS supervisor manages all transactions from the OS/390 server.
- The PDM MVS remote access server handles requests from remote clients and submits them to the PDM MVS subsystem.
- The PDM MVS communication access server manages communications to PDM channel-attached nodes, either directly ESCON-attached or through the IBM InfoSpeed Data Gateway.

4.2 The PDM UNIX or Windows NT Subsystem

On UNIX or Windows NT, PDM has applications, collectively referred to as a PDM Node, that complement the functions performed by the PDM MVS subsystem:

- The PDM node router runs on PDM channel-attached nodes only and manages communications across the channel to the OS/390 servers.
- The PDM Node transfer task is an application running on each PDM channel-attached or remote node. It is started by the MVS PDM transfer task and manages a single data transfer task.
- PDM can also be installed on network-attached UNIX or Windows NT platforms and communicate with the OS/390 server through a channel-attached PDM node.
- A PDM FTP network server can also be running on PDM channel-attached nodes to serve non-PDM nodes (that is TCP/IP FTP clients).

4.3 User Interfaces

As shown in Figure 19, IBM exPress Data Mover offers several interfaces to copyr data to and from S/390 servers and UNIX or Windows NT servers:

- The PDM File Transfer Client, referred to as either the MVS-File-Transfer-Client or the Node File Transfer Client, includes FTP-like commands that can be used interactively or in a batch (script) mode
- The PDM ISPF Client, which includes standard ISPF panels
- The PDM-SubSystem-Interface (SSI), an OS/390 Job Control Language (JCL) interface also supported in dynamic-allocation
- The PDM-REXX-Interface (not shown in Figure 19), which offers a programming interface in OS/390.

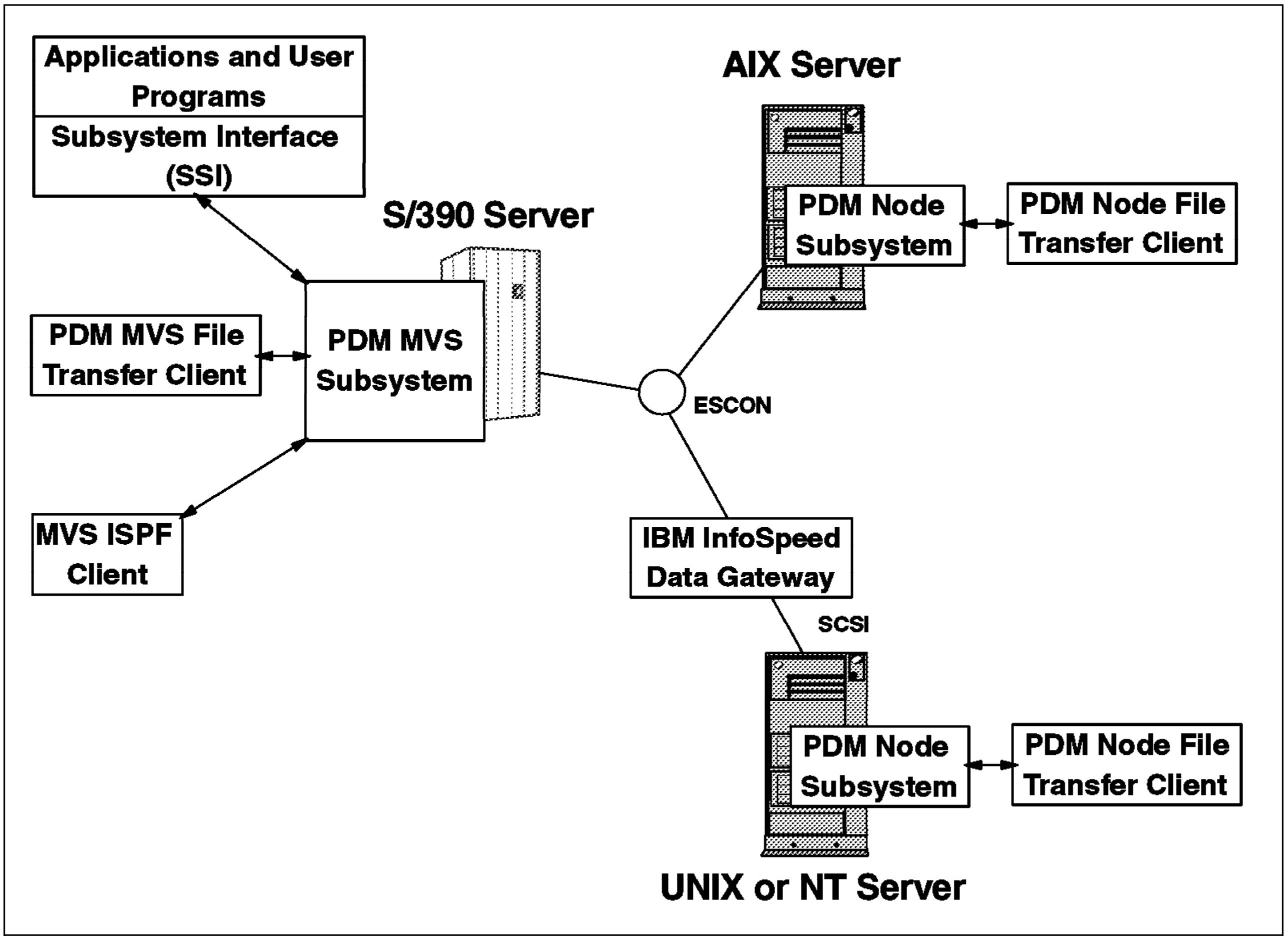

Applications and User Programs
Subsystem Interface (SSI)
S/390 Server
PDM MVS File Transfer Client
PDM MVS Subsystem
MVS ISPF Client
AIX Server
PDM Node Subsystem
PDM Node File Transfer Client
ESCON
IBM InfoSpeed Data Gateway
SCSI
PDM Node Subsystem
PDM Node File Transfer Client
UNIX or NT Server

Figure 19. PDM User Interface Examples

4.3.1 PDM File (MVS or Node) Transfer Client

The PDM File Transfer Client (either MVS or Node) has an FTP-like interface offering standard commands such as open/close, account, ascii/binary, get/put, and cd/lcd. The commands are described in detail in *PDM User's Guide and Reference*, GA32-0388. Other commands are also available and provide additional processing capabilities such as query, sync/nosync, serial/parallel, getx/putx, ldelete, echo, and notify. These commands are specific to PDM and are available with both the MVS and the Node Transfer Client. If PDM is not installed on a UNIX or Windows NT platform, you need PDM installed on a channel-attached node with the PDM FTP Network server configured to act as a gateway between the end-user platform PDM MVS server.

The File Transfer Client can be invoked interactively (TSO on MVS, command line on a node) or in batch mode (MVS native batch or UNIX native script or Windows NT DOS-like *.bat* file).

Using the File Transfer Client (MVS or Node), you can allocate new sequential data sets on OS/390, with or without SMS parameters (MGMTCLAS, STORCLAS, DATACLAS), and UNIX or Windows NT file structures. In addition, new MVS PDS or PDSE members can be created, as well as new versions of GDG data sets.

Use of the getx/putx commands enable you to specify the source or destination of your data to an executable command or script file on a UNIX or Windows NT platform.

The File Transfer Client commands and parameters are described in detail in *PDM User's Guide and Reference*, GA32-0388. Some samples of how to use them are given in Chapter 10, "Using MVS PDM Clients" on page 105.

4.3.2 PDM ISPF Client

The PDM ISPF client is available on the OS/390 server only and consists of standard ISPF panels with fill-in-the-blanks fields for recieving or sending data from or to a UNIX or Windows NT PDM Node. In addition, the ISPF client can monitor and control the PDM system through an Event Manager panel.

4.3.3 PDM Subsystem Interface

The PDM Subsystem Interface (SSI) is available on the OS/390 server only. It enables you to use existing OS/390 applications, that have been developed to use BSAM or QSAM access methods, without modification, to receive or send data to PDM UNIX or Windows NT file structures or executable applications. This is done by modify existing JCL to point to the PDM SSI and parameterize the connection to the UNIX or Windows NT platform.

4.3.4 PDM REXX Interface

The PDM REXX is available on the OS/390 server only and offers a programming interface to control transfers of data (sending or recieving) via the PDM system.

For more information on the REXX interface, see 10.1.3, "Using the PDM REXX Interface" on page 123.

4.4 The IBM exPress Data Mover Architecture

The IBM exPress Data Mover is client/server based software, where three categories of resources can be distinguished:

- The OS/390 PDM server software and client interfaces. The two resources described above are included in the same software package (IBM exPress Data Mover Version 1 for OS/390 platforms) and need to be installed on each OS/390 server.
- The UNIX or NT channel-attached or ESCON-attached (currently AIX only) PDM client.

- The Network-attached PDM client.

Note: The UNIX or Windows NT channel-attached or ESCON-attached PDM node and the network-attached PDM node use the same software (IBM exPress Data Mover Version 1 for UNIX and Windows NT platforms), only configured differently.

Each of these channel-attached resources uses separate communication and data transfer paths between the OS/390 PDM server software and the client software on a UNIX or Windows NT channel-attached node (or ESCON-attached for AIX). Two communications paths are defined on each channel-attached node. Several data transfer tasks can be defined in parallel.

UNIX or Windows NT servers without the PDM software installed can also communicate and transfer data with an OS/390 PDM server using the standard FTP TCP/IP interface. For this purpose, one of the channel-attached or UNIX or Windows NT servers or ESCON-attached AIX server is defined as the PDM FTP Network server, and a specific daemon (or service) has to be started. This function is described in Chapter 9, “Configure a PDM FTP Server for Non-PDM Nodes” on page 103.

Now that we have described both the PDM MVS subsystem and PDM UNIX or Windows NT software, we look next at what their communication is based on and how commands and data are transferred across the ESCON channel or the gateway.

If connecting through a IBM InfoSpeed Data Gateway the communication is based on a mapping of SCSI target IDs/LUN #s and ESCON device addresses. If going through an ESCON adapter the communications is based on a mapping of device-addresses to local device numbers and fiber definitions.

4.4.1 SCSI Target IDs/LUN#s

When defining the SCSI connections to the UNIX or Windows NT server using a SCSI-2 F/W adapter, the IBM InfoSpeed Data Gateway appears as one or more SCSI Target IDs, and up to eight LUNs each. One SCSI connection to the gateway from a directly-attached UNIX or Windows NT platform equals one target ID composed of eight LUNs (numbered 0 to 7). If multiple UNIX or Windows NT platforms are connected to an IBM InfoSpeed Data Gateway Multiplexor, the the SCSI connection between the multiplexor and the gateway equals a target ID and associated LUNs for each platform connected to the multiplexor. The IBM InfoSpeed Data Gateway uses the first two LUNs as control paths, and the next, up to six, LUNs as data paths.

One transfer request uses one LUN. Up to six parallel transfers are thus allowed.

The SCSI target IDs/LUN#s are mapped within the IBM InfoSpeed Data Gateway to the defined ESCON device numbers that have been configured on the OS/390 server and must equal in number, that is a one-to-one mapping of ESCON device number to SCSI Target ID/LUN#.

4.4.2 ESCON Subchannels

When using an ESCON subchannel in OS/390, you can define up to 32 ESCON device addresses to an IBM InfoSpeed Data Gateway, per ESCON connection. The determining factor when defining these addresses is the number of SCSI connections used to attach the UNIX and Windows NT platforms. The PDM ESCON to LUN architecture is based on a one-to-one mapping. The first two ESCON subchannels are used as control paths, and the next (up to) six are used as data paths. For example, if there is only going to be one UNIX or Windows NT server attached on the SCSI side of the gateway, and only one ESCON channel is attached, only define a maximum of eight devices addresses.

4.4.3 Multiple Connections to a Single Server

PDM supports multiple interfaces to a single UNIX or Windows NT server. For AIX, this could be multiple direct ESCON attachments or SCSI attachments using the gatway or combinations thereof. For all UNIX or Windows NT platforms this could also be multiple SCSI attachments. PDM uses only two control paths per node regardless of the number of connections. All other paths can be used for data transfers. For example, if two SCSI interfaces are connected to the same server, two LUNs are used as control paths and remaining six LUNs are used as data-paths. The total complement of the eight LUNs on the other interface (SCSI Target ID) are used as data paths, thus giving a total of 14 data paths to this particular server.

4.5 The IBM exPress Data Mover and Its Paths on OS/390

The primary component is the PDM MVS subsystem, which runs as a started task.

The OS/390 systems administrator builds a configuration table, which defines the channel-attached nodes on the SCSI side of the IBM InfoSpeed Data Gateway or the ESCON-attached AIX nodes, as well as network-attached PDM nodes.

The systems administrator then defines:

- The starting subchannel address that will be used for communications to each channel-attached node
- The number of subchannel addresses that will be available for data transfer operations (which also defines the number of concurrent transfers allowed) to each channel-attached node.

At startup, the PDM MVS subsystem automatically starts PDM MVS remote access server, PDM MVS communication access server, and a pool of (specified by the system administrator in the MVS PDM configuration table) MVS PDM transfer tasks. When a transaction (transfer) is requested, it is handed to an available transfer task and assigned a transfer path for the duration of the transaction. When a transaction completes, the path and the task are returned to the pool.

On the UNIX or Windows NT server side, the PDM node router must be active to make a complete connection.

For each transfer request the flow is:

- A MVS PDM transfer task starts a PDM UNIX or Windows NT transfer task.

- For each concurrent transfer operation, an instance of the PDM UNIX or Windows NT transfer task is started, unless the maximum number of transfer tasks are in use or unless the total number of transfer subchannels has been used.

Note: Transactions are queued for a number of reasons. See ″Why Your Transaction May Be Queued″ in *PDM User′s Guide and Reference*, GA32-0388. You may run out of tasks or you may run out of paths. These are totally independent.

4.6 The IBM exPress Data Mover Used on UNIX or Windows NT

In this section, we describe the communication-and data transfer paths used by IBM exPress Data Mover on UNIX or Windows NT servers.

The available interfaces are the PDM Node File Transfer Client interface (for PDM nodes) or the PDM Network FTP Server, running on a PDM channel-attached node, which provides an TCP/IP FTP interface for non-PDM nodes.

4.6.1 PDM Nodes

In the case of a PDM channel-attached node, the PDM transfer task copies the data directly to or from the file on the node. The PDM-Node-Transfer-Client submits requests and checks on the status through the PDM MVS subsystem. This client can also submit requests that pipe data directly to or from node executables.

Data can be copied to or from a PDM network-attached node. In this case a process called dmeserv is initialized on the channel-attached node and a PDM transfer task is initialized on the network-attached node. The data is then copied by PDM over a TCP/IP-socket to the network-attached node.

4.6.2 Non-PDM Node

The protocol responsible for all data transfers beyond the PDM channel-attached node is TCP/IP. The steps are these:

1. A TCP/IP FTP client ″ftp″s to the PDM channel-attached node that has the PDM FTP network server configured.
2. The PDM FTP server communicates internally to submit the request through the PDM node router.
3. The communication proceeds to PDM MVS which verifies access authority. PDM MVS then selects an MVS PDM transfer task. That action causes an PDM transfer task on the platform to start and be linked to the PDM FTP network server.
4. When data moves, the PDM transfer task moves the data directly to the client.

Note: Non-PDM nodes can also be accessed via all of the MVS clients. In this case, an FTP client if brought up on the channel-attached node which connects to the TCP/IP FTP server on the downstream node.

Chapter 5. Install and Configure IBM InfoSpeed

This chapter explains how to install and configure IBM InfoSpeed using the Data Gateway and Multiplexors.

5.1 IBM InfoSpeed Data Gateway installation

As shown on Figure 20, the IBM InfoSpeed Data Gateway comprises one or two ESCON ports (ESCON 1 and ESCON 2), respectively corresponding to two SCSI ports (PCI 1 and PCI-2), one or two power connections, one or two On/Off switch button, two serial ports (1 and 2) and one Ethernet port.

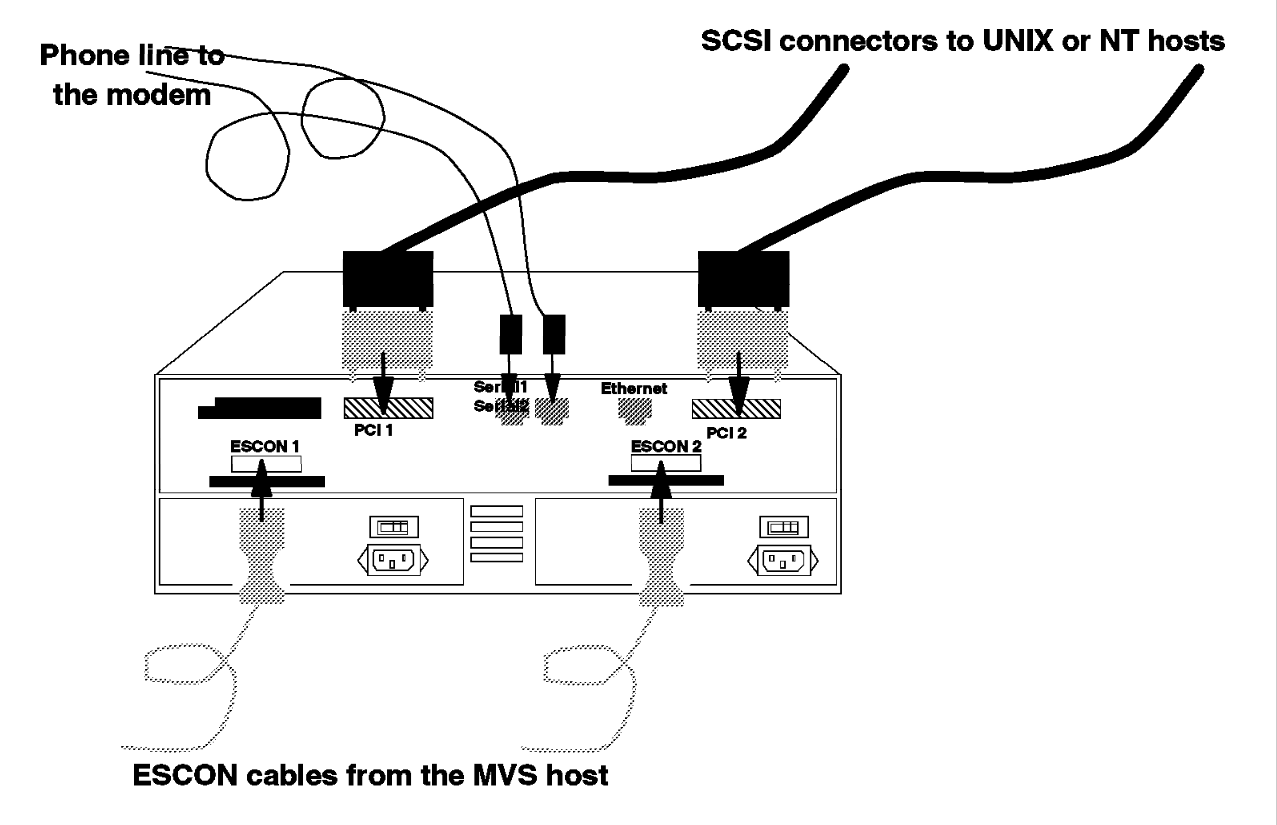


Figure 20. IBM InfoSpeed Data Gateway Connections

When configuring the IBM InfoSpeed Data Gateway only, the connections should be made as described in Figure 20. The Serial 1 is used for diagnosis of the ESCON ports. The Serial 2 port is used for diagnosis of the SCSI ports, to load a new configuration, or to make any changes in the gateway configuration. To download microcode on the IBM InfoSpeed Data Gateway, use the Serial 2 port. Once the IBM InfoSpeed Data Gateway is configured and ready to work, you can remove (not recommended) the phone connection.

Make sure to connect the proper communication cables to the right place. To make a direct IBM InfoSpeed Data Gateway to modem connection, use only the AA Model adapter (P/N 21490054 Rev A).

5.2 IBM InfoSpeed Multiplexor Installation

Figure 21 shows an IBM InfoSpeed installation that includes one IBM InfoSpeed Data Gateway and one IBM InfoSpeed Multiplexor. The IBM InfoSpeed Multiplexor has eight slots, numbered from 1 to 8. Slot 1 is used for the processor card. Slot 2 is used to connect the IBM InfoSpeed Multiplexor to the IBM InfoSpeed Data Gateway. Slots 3 to 8 attach the UNIX or Windows NT servers.

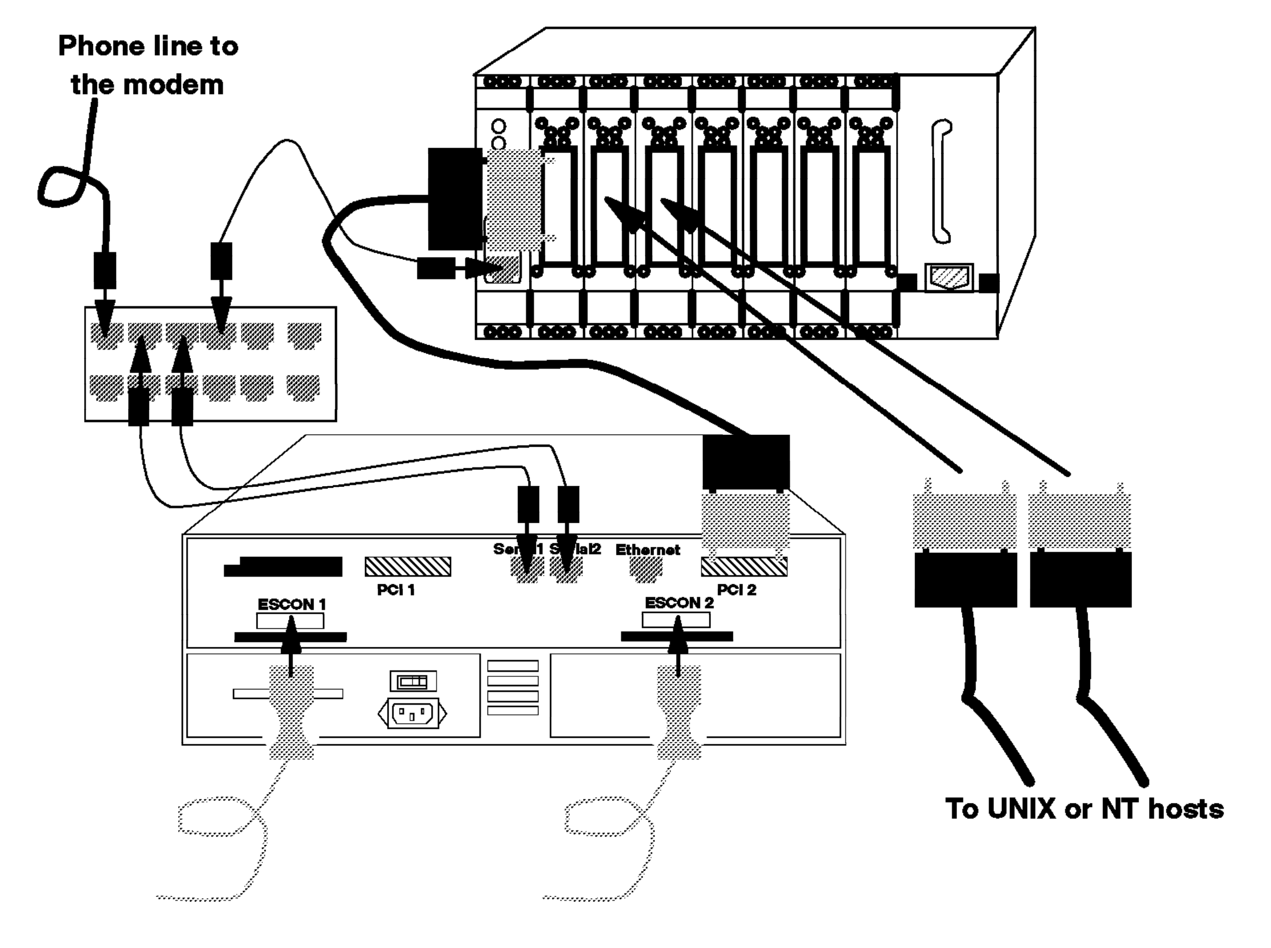


Figure 21. IBM InfoSpeed Multiplexor Connections

If the Multiplexor is used, then the modem needs to be connected to the Serial 2 port.

Such installation with both a IBM InfoSpeed Data Gateway and one or two IBM InfoSpeed Multiplexors requires a modem-switch, and specific adapters (P/N 21490030 Rev B (H)) between the switch and the modem.

The order in which cables should be attached, as shown in Figure 21 and Figure 22 on page 33, to the modem switch ports is:

1. To the modem itself
2. To port Serial-1 on the Gateway
3. To port Serial 2 on the Gateway
4. To port Serial 2 on the first Multiplexor

5. To port Serial-2 on the second Multiplexor
6. To port Serial 1 on a second Gateway
7. ... repeat as for the first Gateway

If you add an IBM InfoSpeed Multiplexor to an already configured IBM InfoSpeed Data Gateway, the configuration in the IBM InfoSpeed Data Gateway has to be updated. Call your IBM CE, and provide him or her with the information required, as shown in Table 2 on page 37 and Table 3 on page 38.

5.3 IBM InfoSpeed Ordering Information

When an InfoSpeed configuration is ordered, you must check that the configuration has the right features, as shown in Figure 22

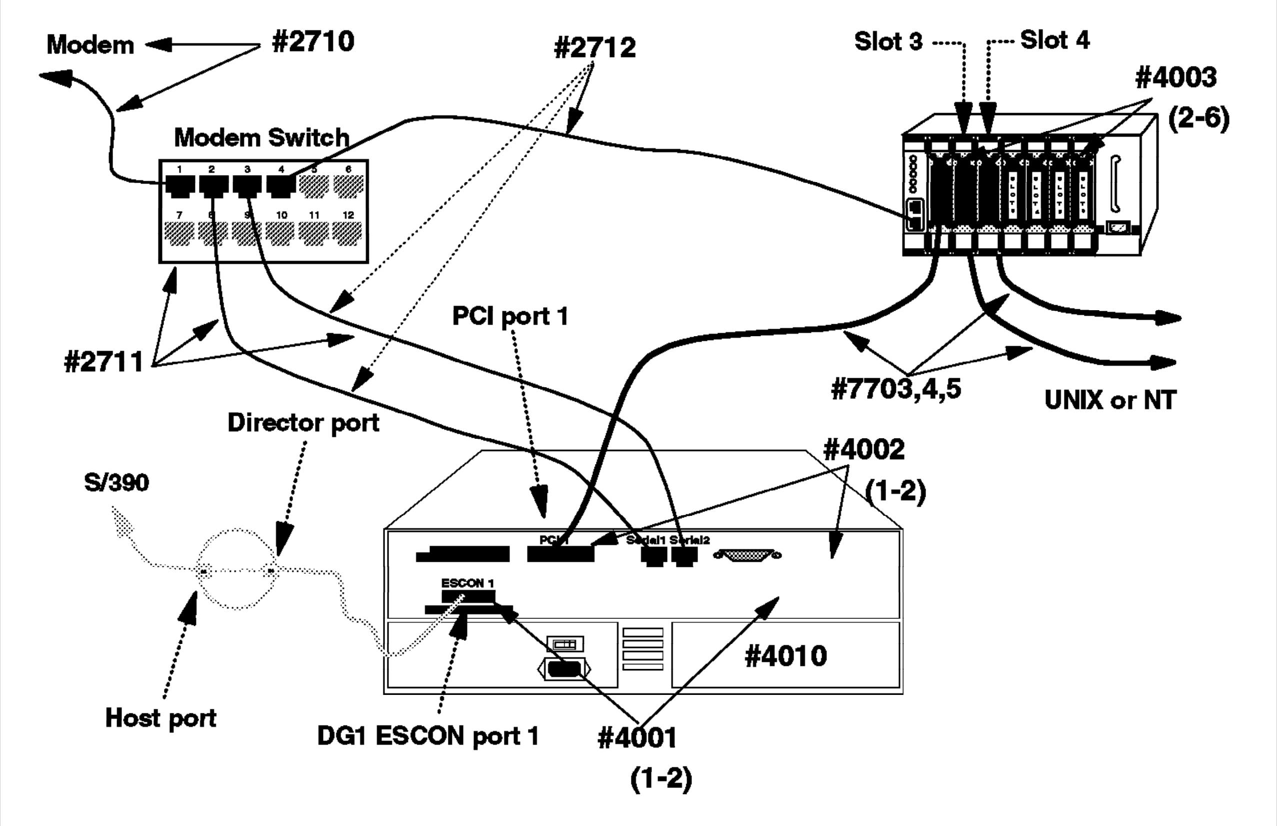


Figure 22. IBM InfoSpeed Data Gateway and IBM InfoSpeed Multiplexor Cabling and Features

The following features are available for the IBM InfoSpeed Data Gateway and IBM InfoSpeed Multiplexor:

- ***Remote support***
  - Feature ***2710***, a cable and connectors for one connection to an IBM-supplied modem
  - Feature ***2711***, a modem switch, cables, and connectors
  - Feature ***2712***, an additional cable and connector

    The first IBM InfoSpeed Data Gateway ordered should have both Feature 2710 and Feature 2711 specified. The second IBM InfoSpeed Data

Gateway should should have Feature 2712 specified twice, as only the connection cables are needed and no modem switch.

- ***Gateway connections***
  - Feature ***4001***, an ESCON port; order one or two
  - Feature ***4002***, an SCSI port, order one or two
  - Feature ***4010***, second power supply, optional
- ***Multiplexor connections***
  - Feature ***4003***, SCSI slot; order two to six per multiplexor
- ***Additonal specifications for the Gateway and the Muliplexor***
  - Feature ***5000***, free standing
  - Feature ***5001***, rack mounted
- ***SCSI cables***
  - Feature ***7703***, 12 meters
  - Feature ***7704***, 18 meters
  - Feature ***7705***, 25 meters

  You should order one SCSI external differential cable for each SCSI port installed on the Gateway and one SCSI external differential cable for each of slot 3 through 8 installed on the Multiplexor. Terminators have to be ordered accordingly.

5.4 IBM InfoSpeed Configuration

An InfoSpeed implementation project can last for many weeks. Duration does not depend on the complexity of the products, but on the distribution of the information needed. In many companies or enterprises, the skill and management of the S/390 systems and the different UNIX and Windows NT systems are spread across many departments and persons. Another aspect is that over the years, those working in a S/390 department as well as those working in the UNIX environment have each adopted a certain image of the other group. For this reason, and because the groups use different terminology, information gathering may be difficult.

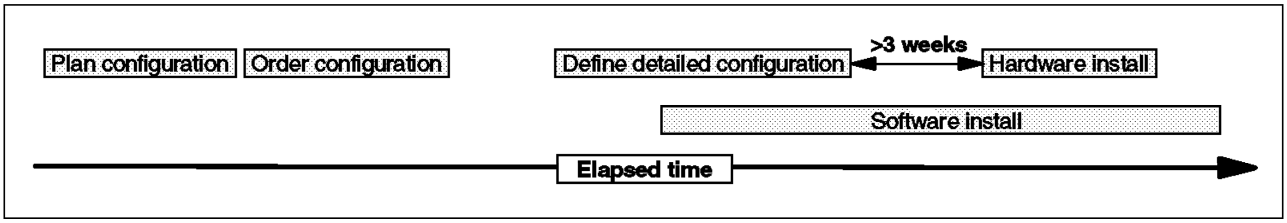


Figure 23. Sample InfoSpeed Installation Project

An installation project for an InfoSpeed solution is basically split into five parts, as shown in Figure 23.

- ***Plan configuration***. As described in Chapter 3, "IBM InfoSpeed: Configuration Alternatives" on page 19, many alternatives are available when selecting an InfoSpeed configuration. Requirements such as the bandwidth needed for each attached UNIX or Windows NT server, the extent of availablity, and the amount of parallelism needed have to be determined.

These requirements are needed in order to configure the right number of attachments for each server as well as how many ESCON channels will be required on the S/390 hosts.

The physical location of the servers to be attached is also a parameter, because the SCSI interface is limited to 25 meters. If the server is to be attached to an IBM InfoSpeed Multiplexor, the SCSI cable between the Multiplexor and the IBM InfoSpeed Data Gateway extends the distance between the Gateway and the attached server to 50 meters. The maximum distance from the Gateway to the S/390 server or the ESCON Director is 3 kilometers.

You also have to consider either using an IBM InfoSpeed Data Gateway or a direct ESCON attach for platforms having this choice of attachment.

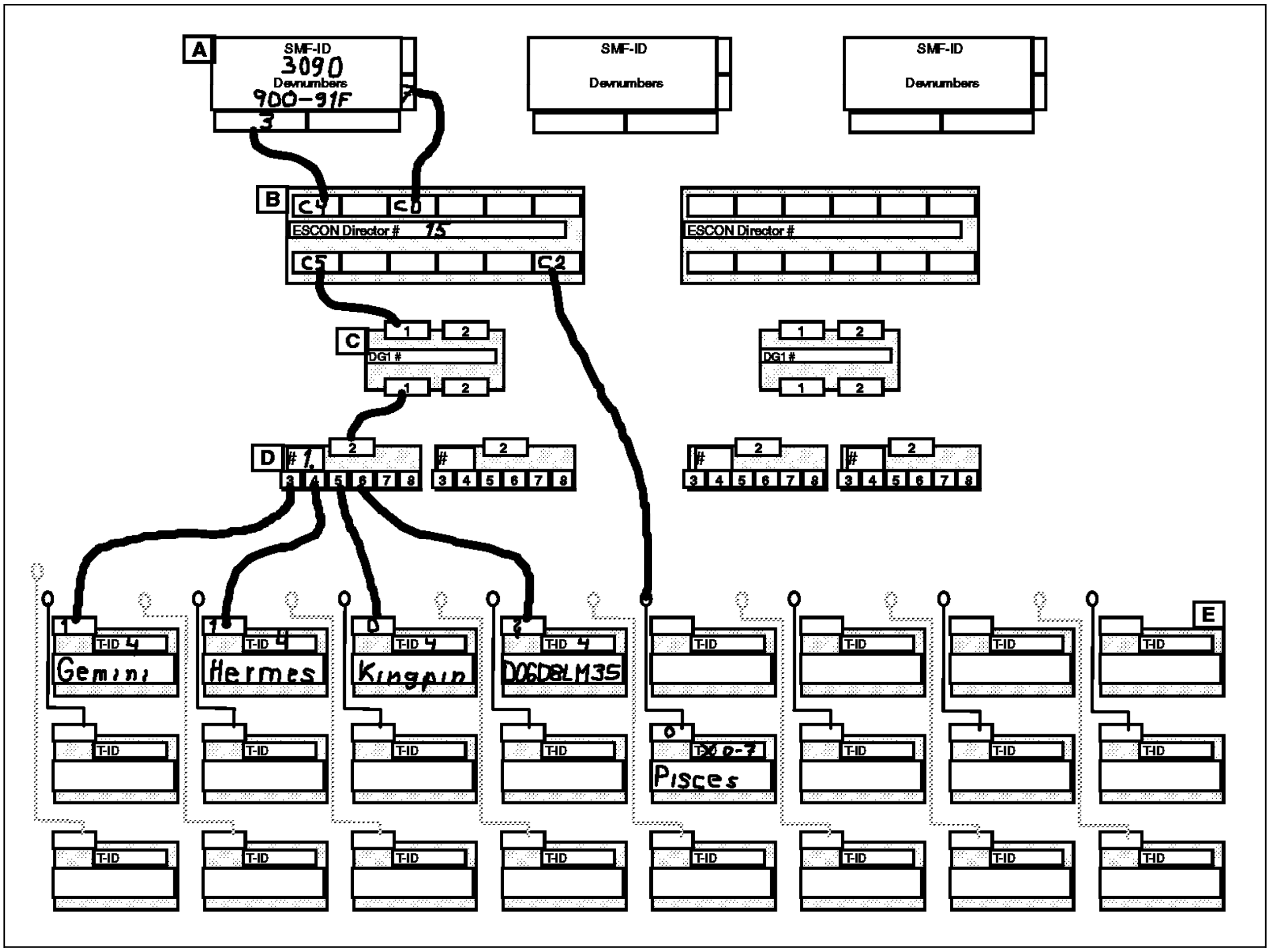


Figure 24. Sample InfoSpeed Configuration Chart

Figure 24 shows a diagram of a sample configuration drawn to assist the ordering of the configuration. To use a similar diagram to document your configuration, see A.1.1, “Gateway Configuration Chart” on page 167.

- ***Order configuration***. An InfoSpeed order does not consist of merely one or more IBM InfoSpeed Data Gateway and IBM InfoSpeed Multiplexor boxes. For each box, check that the features described in 5.3, “IBM InfoSpeed Ordering Information” on page 33 have been included. Additionally, required ESCON and SCSI cables have to be ordered. Installed ESCON Directors may need additional ports to attach the InfoSpeed configuration.

On all servers to be attached, the correct adapter must be ordered (if not already installed).

When ordering the software parts for the configuration, IBM exPress Data Mover for OS/390, UNIX, or Windows NT, remember that they can be installed before the hardware is delivered.

- ***Define detailed configuration***. Frequently requirements change between the time a configuration is ordered and the time it is actually delivered. As the IBM InfoSpeed Data Gateway will be configured at manufacturing time, the documents describing it must be faxed to the plant site well before delivery. Before the information is documented, revisit the configuration diagram created before the InfoSpeed configuration was ordered. Hopefully you find that you have ordered the right equipment and the only change needed is in the way the configuration should be attached. For example, the ports needed in an ESCON director may no longer be available, and a new one may have to be selected.

  After the configuration has been checked, enter the information required by the documentation in Appendix A, “Configuration Worksheets” on page 167. This documentation is also available in *IBM InfoSpeed 9398 Data Gateway Models DG1 and MX1 Planning Guide*,GA32-0395. When the forms have been completed they should be given to the IBM CE or SE who will fax them to the manufacturing site.

  In Table 2 on page 37 through Table 5 on page 40, we show sample specifications for the configuration drawn in Figure 24 on page 35 (and later in Figure 27 on page 59) documented in the required matrices. The columns are described in A.1.2.3, “Information Matrix Field Definitions” on page 177 and A.1.3.2, “Direct ESCON Adapter Configuration Matrices Field Definitions” on page 183.

Table 2. ESCON Port 1 MVS Host Information; Customer name: TUCSON RESIDENCY

DG1 No. (1)	MVS Path (2)	SMF Host ID (3)	HCD LPAR Num. (4)	HCD CHPID (5)	HCD CHPID Type (6)	HCD CHPID Oper Mode (7)	HCD CUADD (8)	HCD Dev. Model (9)	HCD Device Number (10)	HCD Unit Addr (11)	VM/ESA Virtual Device Address (12)	ESCD Num. (13)	ESCD S/390 Port (14)	ESCD DG1 Port (15)
TBD	Cntrl Path 1	***3090***	***7***	***3***	CNC	***REC***	0	SCTC	***910***	***00***	***TBD***	***15***	***C4***	***C5***
	Cntrl Path 2				CNC			SCTC	***911***		***TBD***			
	Data Path 1				CNC			SCTC	***912***		***TBD***			
	Data Path 2				CNC			SCTC	***913***		***TBD***			
TBD	Cntrl Path 1	***3090***	***7***	***3***	CNC	***REC***	0	SCTC	***914***	***00***	***TBD***	***15***	***C4***	***C5***
	Cntrl Path 2				CNC			SCTC	***915***		***TBD***			
	Data Path 1				CNC			SCTC	***916***		***TBD***			
	Data Path 2				CNC			SCTC	***917***		***TBD***			
TBD	Cntrl Path 1	***3090***	***7***	***3***	CNC	***REC***	0	SCTC	***918***	***00***	***TBD***	***15***	***C4***	***C5***
	Cntrl Path 2				CNC			SCTC	***919***		***TBD***			
	Data Path 1				CNC			SCTC	***91A***		***TBD***			
	Data Path 2				CNC			SCTC	***91B***		***TBD***			
TBD	Cntrl Path 1	***3090***	***7***	***3***	CNC	***REC***	0	SCTC	***91C***	***00***	***TBD***	***15***	***C4***	***C5***
	Cntrl Path 2				CNC			SCTC	***91D***		***TBD***			
	Data Path 1				CNC			SCTC	***91E***		***TBD***			
	Data Path 2				CNC			SCTC	***91F***		***TBD***			

Table 3. SCSI Information for ESCON Port 1; Customer name: TUCSON RESIDENCY

SCSI Port PCI-1 or PCI-2 (16)	MX1 Name & Slot # (17)	UNIX/NT Server Name (18)	SP2 I/F Name (19)	H/W Platform (20)	Oper. System (21)	O/S Vers. (22)	PDM FTP Ntwk. Server (23)	PDM FTP Ntwk. Server I/F Name (24)	PDM Path (25)	SCSI Adapter Type (26)	SCSI Target ID (27)	SCSI LUN No. (28)	Vendor ID (29)	Product ID (30)
1	1-3	gemini	N/A	RS/6000	AIX	4.2.0	Y or N Y	gemini tucson.ibm.com	Cntrl Path 1	F&W DF	4	0	Other SCSI	ost
									Cntrl Path 2			1		
									Data Path 1			2		
									Data Path 2			3		
1	1-4	hermes	N/A	HP/9000 D270	HP-UX	10.20	Y	hermes tucson.ibm.com	Cntrl Path 1	F&W DF	4	0	Exabyte	8500
									Cntrl Path 2			1		
									Data Path 1			2		
									Data Path 2			3		
1	1-5	kingpin	N/A	TBD	Solaris	2.5	Y	kingpin tucson.ibm.com	Cntrl Path 1	F&W DF	4	0	Exabyte	8500
									Cntrl Path 2			1		
									Data Path 1			2		
									Data Path 2			3		
1	1-6	D06D8LM35	N/A	TBD	Win NT	4.0	Y	D06D8LM35 tucson.ibm.com	Cntrl Path 1	F&W DF	4	0	Exabyte	8500
									Cntrl Path 2			1		
									Data Path 1			2		
									Data Path 2			3		

Table 4. (pisces) Direct ESCON Adapter UNIX Information; Customer name: TUCSON RESIDENCY*

ESCON Adapter Number (1)	UNIX/NT Server Name (2)	SP2 I/F Name (3)	H/W Platform (4)	Oper. System (5)	O/S Vers. (6)	PDM FTP Ntwk. Server (7)	PDM FTP Ntwk. Server I/F Name (8)
0	***pisces***	***N/A***	***RS6000***	***AIX***	***4.2***	Y or N ***Yes***	***pisces*** ***tucson.ibm.com***

Table 5. (pisces) Direct ESCON Adapter (Part 1 of _1) S/390 Information; Customer name: TUCSON RESIDENCY*

Local Address (9)	MVS Path (10)	SMF Host ID (11)	HCD LPAR Num. (12)	HCD CHPID (13)	HCD CHPID Type (14)	HCD CHPID Oper Mode (15)	HCD CUADD (16)	HCD Dev. Model (17)	HCD Device Number (18)	HCD Unit Addr (19)	VM/ESA Virtual Device Address (20)	ESCD Num. (21)	ESCD S/390 Port (22)	ESCD Adapter Port (23)
0	Cntrl Path 1	***3090***	***7***	***7***	CNC	***SHR***	***0***	RS6K	***900***	***00***	***TBD***	***15***	***C0***	***C2***
1	Cntrl Path 2				CNC			RS6K	***901***		***TBD***			
2	Data Path 1				CNC			RS6K	***902***		***TBD***			
3	Data Path 2				CNC			RS6K	***903***		***TBD***			
4	Data Path 3				CNC			RS6K	***904***		***TBD***			
5	Data Path 4				CNC			RS6K	***905***		***TBD***			
6	Data Path 5				CNC			RS6K	***906***		***TBD***			
7	Data Path 6				CNC			RS6K	***907***		***TBD***			
8	Data Path 7				CNC			RS6K	***TBD***					
9	Data Path 8				CNC			RS6K	***TBD***					
10	Data Path 9				CNC			RS6K	***TBD***					
11	Data Path 10				CNC			RS6K	***TBD***					
12	Data Path 11				CNC			RS6K	***TBD***					
13	Data Path 12				CNC			RS6K	***TBD***					
14	Data Path 13				CNC			RS6K	***TBD***					
15	Data Path 14				CNC			RS6K	***TBD***					

- ***Hardware install***. Before the hardware is attached, it is advisable that you complete step 6.2.1, “Define Devices to MVS IOCP or to HCD” on page 44 and step 6.2.2, “Disable Missing Interrupt Handler” on page 45.

  Your IBM InfoSpeed Data Gateway and IBM InfoSpeed Multiplexor should arrive already configured with the parameters you gave to your IBM CE or SE when you ordered the gateway. If they do, then just push the power switch and step forward to Chapter 6, “PDM MVS Installation and Configuration” on page 43

  If this is not the case, or if you changed your mind as to how the Gateway should be configured, then call your IBM hardware CE, and provide him with the new information. The configuration loaded into the Gateway at manufacturing time can be updated remotely.

  Once you have sent the required informations, the IBM CE will call the configuration center, which will download the configuration directly to your IBM InfoSpeed Data Gateway through the phone line.

  If you can′t allow a direct phone line connection from the outside into the Gateway, then your IBM CE will make a specific appointment with you to configure the IBM InfoSpeed Data Gateway on site.

  In any case, ***don′t try to connect and configure the IBM InfoSpeed Data Gateway by yourself unless you have been trained for that***.

- ***Software install***. The InfoSpeed hardware need not to be installed in order to install the software, IBM exPress Data Mover, on the different platforms. The details on how to install the software are detailed in Chapter 6, “PDM MVS Installation and Configuration” on page 43 through Chapter 9, “Configure a PDM FTP Server for Non-PDM Nodes” on page 103.

Chapter 6. PDM MVS Installation and Configuration

This chapter provides instructions for installing and customizing the IBM exPress Data Mover (PDM) software and PDM MVS clients on the OS/390 server.

Installation of PDM on the MVS system is a process comprising these actions:

- Defining physical input/output (I/O) devices
- Defining an installation-specific high-level qualifier for all PDM data sets
- Editing members in the SYS1.PARMLIB dataset
- Installing the SMP/E distribution tape
- Editing and executing jobs shipped with PDM
- Defining DMES as a subsystem to MVS
- Disabling the missing interrupt handler for all devices
- Loading MVS libraries
- Linking MVS libraries
- Defining SDMELINK as an authorized library list
- Allocating and initializing QUEUE
- Allocating and initializing HDETABLE

6.1 MVS Software Prerequisites

Before you install PDM, make sure you have the following:

- MVS/ESA at Release SP 5.2.2 or higher, or OS/390 Version 1 Release 3 or higher
- VS COBOL II Run Time Library or LE/370

  Note: This library is required only if PDM MVS FTP Clients are used.
- Catalog-Search-Interface (5799-CSI) or DFSMS/MVS Version 1 Release 4

  Note: Required for MVS directory listings. If not installed, the MVS directory listing function of DME is disabled.
- Resource Access Control Facility (RACF) or a similar security product supporting the SAF interface may be required. For more information, see Appendix B, “OS/390 Security Processing in IBM exPress Data Mover” on page 185.

Note: Always check *Program Directory for IBM InfoSpeed exPress Data Mover*, GI10-0542, for latest information.

6.2 PDM MVS Subsystem Installation Steps

The following steps are required before the PDM MVS subsystem is installed:

1. Define devices to MVS (using IOCP or HCD).
2. Define DMES as a subsystem.
3. Load the PDM files from the PDM Distribution Tape.

4. Disable the Missing Interrupt Handler.

6.2.1 Define Devices to MVS IOCP or to HCD

All PDM channel-attached nodes (IBM Gateway, integrated channel adapter) must be defined to the MVS IOCP or HCD:

CHPID The IBM InfoSpeed Data Gateway and a direct ESCON attach should be connected to channels defined as **TYPE=CNC**.

CNTLUNIT The IBM InfoSpeed Data Gateways must be defined as **TYPE=SCTC** control units. A single statement can define several attached systems.

A direct ESCON-attached system must be defined as **TYPE=RS6K** control units. Each system must be defined as a separate control unit.

Note: A new PCI-based adapter is available but has not yet been verified to be used by IBM exPress Data Mover. This new adapter will be specified as a **TYPE=RS6K-2** control unit if APAR OW33336 is installed on the OS/390 system. If the APAR is not installed, use **TYPE=RS6K**.

IODEVICE The IBM InfoSpeed Data Gateway device numbers must be defined as **UNIT=SCTC**. A single statement can define several attached systems.

A direct ESCON-attached system must be defined as **UNIT=RS6K** device numbers. A single statement defines device numbers for one attached system.

For each channel-attached node, define two devices (subchannels) for communications and the rest for data transfer. Define enough data-transfer-devices (subchannels) to support the maximum number of concurrent transfers desired.

For each IBM InfoSpeed Data Gateway used, assign enough device numbers to support the maximum number of concurrent transfers desired for all SCSI-attached nodes, plus two additional device numbers for communication for each node. For example, it two UNIX platforms will be SCSI-attached, you could assign a maximum of 16 device numbers, 12 for data transfer and 4 for control (2 for each UNIX).

The following are typical definitions for attachment of up to four SCSI connections through an IBM InfoSpeed Data Gateway:

```
CHPID       PATH=(7),SWITCH=03,TYPE=CNC

CNTLUNIT    CUNUMBER=900,PATH=(7),UNIT=SCTC,LINK=(CA),
            UNITADD=((00,32))

IODEVICE    ADDRESS=(900,32),UNIT=SCTC,CUNUMBER=(900),STADET=Y
```

If you use the Hardware-Configuration-Definition (HCD) to add the devices, follow the instructions in HCD regarding installation of an SCTC (for the Gateway) or 3088 (for the direct ESCON attachment) using the above example as a guide.

6.2.2 Disable Missing Interrupt Handler

During data transfer to and from MVS tape volumes where an end of volume is reached, a read or write operation to a subchannel may not complete (because of a pending MVS tape mount) within the default time allotted by the MVS Missing Interrupt Handler (MIH). To avoid an error MIH must be disabled for the range of device addresses for each CTC adapter.

For example, to disable the MIH for the range of devices 500 to 51F, edit member IECIOSxx in SYS1.PARMLIB and add the following line:

```
MIH TIME=00:00,DEV=(500-51F)
```

Each range of devices requires a separate line. If member IECIOSxx does not exist, a new member must be created and the following entry added in member IEASYSxx in the library, that refers to this new member IECIOSxx:

```
IOS=xx       (where xx = suffix to IECIOS)
```

If you plan to activate PDM before the next initial program load (IPL), enter a SET IOS=xx command from the console to apply this change.

6.2.3 Define PDM as a Subsystem

Edit member IEFSSNxx in library SYS1.PARMLIB, where xx is the suffix used in the SSN= parameter in the most current IEASYSxx member, and add the following:

```
SUBSYS SUBNAME (DMES)
```

An IPL is required after this step to invoke the definition, but the subsystem can be added dynamically by the following command:

```
SETSSI ADD,S=DMES
```

If the command is entered but the IEFSSNxx member has not been updated with the DMES change, the subsystem will ***not*** be defined after the next MVS system IPL.

IBM recommends that you use the keyword parameter form of the IEFSSNxx parmlib member (as coded above). However, the positional parameter form of the IEFSSNxx parmlib member is still supported.

The primary-PDM-subsystem should always be DMES; for other nonprimary PDM subsystems, any four-character name can be used.

6.2.4 Load the Required Files from the PDM Distribution Tape

To load the distribution-libraries required for PDM, code the JCL shown in Figure 25 on page 46.

```
//JOBNAME  JOB(..........),′PDM′,CLASS=A,MSGCLASS=A,
//           REGION=4M,MSGLEVEL=(1,1),NOTIFY=XXXXXXX
//STEP1    EXEC PGM=IEBCOPY,PARM=′SIZE=250K′
//SYSPRINT DD  SYSOUT=*
//RELF3    DD  DSN=PDM.HPDM110.F3,LABEL=(4,SL),
//             VOL=SER=PDM110,DISP=(OLD,PASS),UNIT=TAPE
//JOBSF    DD  UNIT=SYSDA,DNS=hlq.PDM110.JOBSFILE,DISP=(,CATLG),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160),
//             SPACE=(TRK,(5,5,1))
//SYSIN DD *
  COPY I=RELF3,O=JOBSF
  SELECT MEMBER=DMESETUP
  SELECT MEMBER=DMEDDDEF
  SELECT MEMBER=DMEACCHK
  SELECT MEMBER=DMEACCKM
  SELECT MEMBER=DMEACCUM
  SELECT MEMBER=DMEACPDM
  SELECT MEMBER=DMEAPCHK
  SELECT MEMBER=DMEAPCKM
  SELECT MEMBER=DMEAPCUM
  SELECT MEMBER=DMEAPPDM
  SELECT MEMBER=DMERCCUM
  SELECT MEMBER=DMERCPDM
  SELECT MEMBER=DMERJPDM
/*
//
```

Figure 25. Sample JCL to Copy Install Members from PDM Distribution Tape

These members contain JCL to install the IBM exPress Data Mover software on the MVS system.

Member	**Description**
DMESETUP	Sample job to allocate target and distribution libraries
DMEDDDEF	Sample job to define SMP/E DDDEFs
DMEACCHK	Sample ACCEPT CHECK job for PDM HPDM110
DMEACCKM	Sample ACCEPT CHECK job for PDM cumulative service
DMEACCUM	Sample ACCEPT job for PDM cumulative service
DMEACPDM	Sample ACCEPT job for PDM HPDM110
DMEAPCHK	Sample APPLY CHECK job for PDM HPDM110
DMEAPCKM	Sample APPLY CHECK job for PDM cumulative service
DMEAPCUM	Sample APPLY job for PDM cumulative service
DMEAPPDM	Sample APPLY job for PDM HPDM110
DMERCCUM	Sample RECEIVE job for PDM cumulative service
DMERCPDM	Sample RECEIVE job for PDM HPDM110
DMERJPDM	Sample REJECT job for PDM HPDM110

This process is also described in detail in *Program Directory for IBM InfoSpeed exPress Data Mover*, GI10-0542.

6.2.5 Install the Distribution Tape

Apply the distribution tape using standard JCL for invoking SMP/E, receive, apply and accept the SYSMOD (HPDM110). Sample JCL is provided to help perform all the installation tasks. The SMP/E dialogs can also be used instead of JCL to accomplish the SMP/E installation steps.

6.2.6 Install the Cumulative Service Tape

If you received a service tape, use standard JCL to receive, apply and accept the tape. Refer to the specific instructions included with the service tape.

For detailed instructions on installing PDM on the MVS system and information on system requirements, refer to *Program Directory for IBM InfoSpeed exPress Data Mover*, GI10-0542.

6.2.7 Allocate the PDM Data Sets

Member DMEALLOC, already loaded from the PDM distribution tape allocates all of the PDM data sets. Choose a prefix for all the PDM data sets and edit this member as required and submit.

6.2.8 Adding DDDEFS to SMP/E

Add DDDEFS for the PDM target and distribution libraries. In a previous step, member DMEDDDEF was loaded from the distribution tape. Edit this member for installation dependencies, make sure that the data set names for each of the DDs match what was previously specified in the DMESETUP member. You can install PDM into any SMP/E zone but you will have to change the SET BDY statements to match the names of the target and distribution zones.

6.2.9 Define SDMELINK as an Authorized Library

Add the following line to the PROGxx member of SYS1.PARMLIB (starting in Column 1):

```
APF ADD DSNAME(SDMELINK) VOLUME(vvvvvv)
```

where vvvvvv is the volume serial number of the volume containing the LINKLIB dataset. The VOLUME parameter is unnecessary if the LINKLIB dataset is cataloged.

To activate the change to a running system, issue the following operator command from the console:

```
SET PROG=xx
```

where xx is the index of your PROGxx member in SYS1.PARMLIB.

IBM recommends using the PROGxx member but instead of using this method, you can use the IEAAPFxx member in SYS1.PARMLIB. Add the following line to the IEAAPFxx member:

```
hlq.SDMELINK        vvvvvv
```

where hlq is the high level qualifier of the SDMELINK data set and vvvvvv is the serial number of the volume where the data set resides.

Both methods are described in *OS/390 MVS Initialization and Tuning Reference*, SC28-1752.

6.2.10 Add SDMELPA to the LPALST

Edit member LPALSTxx in SYS1.PARMLIB and add hlq.SDMELPA to the list of data set names. If you do not have on a running system, a product that can refresh LPA, use an IPL to make this change and reply to the message IEA101I with CLPA. Alternatively, you can use the dynamic LPA function. Refer to *OS/390 MVS Initialization and Tuning Reference*, SC28-1752, for more information.

6.2.11 Add SDMELOAD to the LNKLST

For all of the users to access the PDM MVS clients, concatenate SDMELOAD to your LNKLST. Edit member LNKLSTxx in SYS1.PARMLIB and add hlq.SDMELOAD to the list of data set names. Alternatively, you can use the dynamic LNKLST function. Refer to *OS/390 MVS Initialization and Tuning Reference*, SC28-1752, for more information.

6.2.12 Edit Start Procedure

Member DMESYS in SDMEPROC is the start procedure for the PDM subsystem. Edit this member, changing the high level qualifier (hlq in the sample member) to your installation defined high level qualifier for all PDM datasets defined in this procedure.

6.2.13 Concatenate SDMEPROC to the PROCLIB list

We recommend that you concatenate the SDMEPROC procedure library to SYS1.PROCLIB. This is done in the following way:

- JES2 - Concatenate a DD statement for the SDMEPROC data set to the PROC00 DD in the JES2 start procedure in SYS1.PROCLIB.
- JES3 - Add a DD statement IATPLBnn to the JES3 start procedure in SYS1.PROCLIB for the SDMEPROC data set.

Alternatively, you may copy the DMESYS procedure from the SDMEPROC data set into SYS1.PROCLIB.

6.2.14 Initialize the PDM Queue

Member DMEINITQ in SDMESAMP initializes the queue data set (QUEUE).

The PDM Data in Virtual Queue maintains information about pending, active, and completed transactions. We recommend that you use 1000 entries (specified by the ENTRIES= parameter on the EXEC statement in the DMEINITQ job). The system maintains the transaction information until all entries are used. At that point, the entries are reused and overlaid with new transactions. The ISPF Event Manager displays these entries. In addition, all PDM clients support transaction queries.

Edit DMEINITQ in SDMESAMP for installation requirements and submit it.

6.2.15 Define Nodes to PDM

The HDETABLE data set defines all of the configuration parameters to PDM. Use the DMETABLE program (which has been linked into SDMELOAD) to initialize this data set. The DMETABLE program specifies the following:

- Channel-attached nodes. Defines the file transfer connections to the node as well as the communication paths to the node (a node is also referred to as a remote host).

- Network-attached PDM nodes, which are addressable through the network by using the channel-attached node as a gateway. The network nodes must be addressable through a TCP/IP connection to the gateway node.
- FTP use (whether or not FTP is used to contact non-PDM nodes through the channel-attached node).
- SMF accounting (which record types to write when MVS data sets are referenced or changed by PDM).
- Default allocation parameters to use for FTP clients when they create new MVS data sets.
- Maximum number of concurrent transfers PDM can perform.
- Privileged MVS user IDs which are permitted to cancel any transaction and to change the state of transfer tasks (to turn them off for example).

The following is a list of members that contain sample JCL to define nodes to PDM in various configurations. All members are contained in the SDMESAMP library:

DMETBASE This is a basic configuration of a single channel-attached node that has no PDM Clients installed (FTP is used).

DMETONE This is a basic configuration of a single channel-attached node that uses PDM Clients.

DMETTWO This is a configuration of two channel-attached nodes with redundant access through the network.

DMETNET This configuration defines many network-attached PDM nodes accessed through a single channel-attached node.

DMETFTP This configures an FTP server on a single channel-attached node.

To use them, follow these steps:

1. Choose the member that best matches your installation′s configuration, and edit the JCL to configure your nodes.
2. Execute the job prior to bringing the subsystem up. Refer to Section 6.3.1, "The DMETABLE Program" on page 51 for details on how to use the DMETABLE program.

6.2.16 Specifying Data Set Security

The started task for the MVS server only requires authorization to access or write PDM datasets defined in the start procedure. Most installations specify the data set security of the started task through the logon ID of the procedure name. PDM uses third-party calls to confirm the user′s authority to access or alter data sets (RACROUTE, SAF router).

PDM will need authority to issue operator commands.

The way IBM exPress Data Mover does security checking depends on the maintenance level installed. For more information, see Appendix B, "OS/390 Security Processing in IBM exPress Data Mover" on page 185.

If you are planning to allow JES job submissions from remote users through either FTP or PDM Node File Transfer clients, use the SURROGATE class for the started-task logon ID so that it can submit jobs on behalf of other users.

6.2.17 Optional Steps

The necessity of performing any of the following steps depends upon your installation′s needs and which features you ordered.

6.2.17.1 Installing the ISPF Interface

If you are installing the ISPF interface, install the panels for the MVS ISPF client, as follows:

1. Add the PDM ISPF option into a suitable existing ISPF panel definition. For example, to use the primary panel ISR@PRIM, code the following additional lines within the panel definition:

   ```
   % 10 +SCLM          - Software Configuration and Library Manager
   % PD +DMES          - exPress Data Mover
   % S  +SDSF          - Spool Display and Search Facility
   ```

2. In the execution portion of the panel entry, specify the PDM CLIST to execute when PD is typed. Within the PROC entries in the panel member, code the following statement:

   ```
   PD,′CMD(DMESISPF)′
   ```

3. Add SDMECLST to the proper TSO logon procedures in the SYS1.CMDPROC. In the section of the logon procedure where SYSPROC is allocated, add a concatenation for the library SDMECLST.

4. Edit member DMESISPF in library SDMECLST with the appropriate data set names used in the installation of the product from the distribution tape and any other installation dependencies.

6.2.17.2 Edit TSO CLISTs

In SDMECLST you find sample clists to run PDM interactively in TSO. Edit the members with data set names used in the installation and store the result in appropriate library.

6.2.17.3 Editing and Building the Translation Tables

PDM is shipped with hard-coded EBCDIC/ASCII translation tables. You can override these tables or create new ones for your installation. The installation process created two libraries, one for editable tables and one for machine-readable tables. The editable tables shipped with PDM are contained in the SDMEXLTB data set. You can create the machine-readable tables in the XLATTAB data set, which was created by the DMESETUP job.

Use the DMERXLAT job in SDMESAMP as a model to convert any editable table to a machine-readable table. Refer to ″The DMEXLATE Program″ in the *PDM User′s Guide and Reference*, GA32-0388, for a detailed explanation of this process.

6.2.17.4 SMF Considerations

Use the SMF command in the HDETABLE data set to turn on SMF accounting and specify record types to be written when PDM accesses MVS data sets. Member DMESMF in data set SDMEMACS contains an assembler language mapping of the SMF records written by PDM

If SMF records are desired, make sure that the selected record types are not filtered by SMF itself, a user-written exit, or an extract program.

6.2.17.5 Installing the REXX Function Pack

There are three members in SYS1.SAMPLIB that map the TSO parameter modules for the various environments:

Member	Environment
TSOREXX1	IRXPARMS - Batch TSO
TSOREXX2	IRXTSPRM - TSO/E Interactive
TSOREXX3	IRXISPRM - ISPF

Decide which of the above environments (MVS batch, TSO/E, or ISPF) will have access to the PDM REXX interface. Also, decide whether to allow all users or only specific users access to the interface.

The PDM function package can be added as either LOCAL or USER. The following instructions add PDM to the TSO/E environment as a LOCAL function package:

1. Copy member TSOREXX2 to an appropriate assembler source library and give it a name that is easy to remember, such as SYSTSPRM.
2. Edit SYSTSPRM and find label PACKTB_HEADER. This is the start of the function package table. Change the value of PACKTB_LOCAL_TOTAL and PACKTB_LOCAL_USED to 2. After the label PACKTB_NEXT_LOCAL, add the following statement:

   ```
   PACKTB_ENTRY_DME DC CL8'DMEREXX'    /* PDM FUNCTION PACKAGE      */
   ```

3. Save the module and assemble it.
4. Link the module (as load module name IRXTSPRM) into either a logon STEPLIB or JOBLIB if you want to limit PDM to specific users, or into LNKLST if you want to make it generally available. The standard MVS version of this module is in SYS1.LPALIB.

Repeat the above steps for any other environment in which you want PDM to execute.

Refer to *OS/390 TSO/E REXX Reference* for detailed instructions.

6.3 PDM MVS Subsystem Configuration

The InfoSpeed configuration is defined to PDM using the DMETABLE program.

6.3.1 The DMETABLE Program

The DMETABLE program is used to build and maintain the PDM configuration, which includes all of the connections between MVS and remote node.

There are six types of statements, which can be coded in any order:

SMF	defines SMF record types
PRIV	defines privileged users
MAXT	defines maximum number of MVS transfer paths
HDEF	defines MVS transfer paths
CDEF	defines MVS communications tasks and remote access servers
SDEF	defines default site parameters for FTP requests

Table 6 on page 52 and Table 7 on page 54 describe each statement and their corresponding parameters.

Table 6 (Page 1 of 2). DMETABLE Parameter Types and Explanation of Each Parameter

Parameter	Explanation
SMF Statement	
STOR=	Record type to use when creating or updating MVS data sets.
RETR=	Record type to use when an MVS data set is read.
REN=	Record type to use when an MVS data set or PDS member is renamed.
DELE=	Record type to use when an MVS data set or PDS member is deleted.
MAXT Statement	
nnn	Defines the maximum number of transfer paths started. The number of transfer paths equals the number of concurrent transfers that PDM can perform. Each transfer path requires approximately 135 KB of storage below the 16 MB line. A 4 MB region can handle approximately 26 transfer paths. The default is 10 or the number of transfer paths defined, whichever is less. Optional.
PRIV Statement	
(mvsuserid,...)	Defines the MVS user IDs of those users who are able to cancel any request and change the state of MVS transfer paths and communication tasks from the PDM MVS ISPF client. Optional.
HDEF Statement	
label HDEF	The label of the HDEF statement is the name of the node (name obtained using the uname command) for which the data paths is to be defined. The label (node name) can have from 1 through 16 characters. Required.
COMM=	Refers to the label on the ***CDEF statement***, further down in this table, for the communications task serving this node. Required.
SUM=	Specifies whether to checksum the data buffers in both directions for this host. Default value is N.
FTP=	Specifies whether FTP transfers are allowed through this node. Use Y for yes and N for no. PDM uses the FTP protocol to contact nodes that are not defined in DMETABLE, do not have an HDEF statement, or are not defined by the SUBHOST= parameter. The default value is Y.
IODRV=	Defines the I/O driver to use for file transfer between the Transfer task and the PDM node. This should be set to DMEIOCTC for integrated channel-adapter-connected nodes and DMEIOCTS for SCSI-attached nodes. Required.
OS=	Defines the operating system type for this node. Allowable values are UNIX and NT. The default is UNIX.
UNIT=	Defines the starting device number for file transfer to and from this node and the number of consecutive additional device numbers to assign. For example, UNIT=(502,10) defines ten units from 502 through 50B. The number of device numbers assigned equals the number of concurrent transfers and thus the bandwidth available on the channel for file transfers. To assign a single device number, do not code parentheses (example UNIT=502). Code device number as either 3 or 4 hex digits. Required.

Table 6 (Page 2 of 2). DMETABLE Parameter Types and Explanation of Each Parameter

Parameter	Explanation
SUBHOST=	Defines PDM nodes that are downstream of this channel-attached node and network-addressable (using TCP/IP) from this node. Use parentheses to specify a list of names, which are then comma (,) delimited. Optional.
CDEF Statement	
label CDEF	The label is the name of the communications task for a specific node. This name is specified on the HDEF statement, parameter COMM=. See ***HDEF Statement*** earlier in this table. The label name can have up to eight characters. Required.
TYPE=	Specifies the type of communications protocol to use. The only currently supported value for this is CTC.
QSIZE=	Specifies the size of the communications queue. The default is 100 entries, which is sufficient unless many downstream nodes are used (SUBHOST or FTP=Y). A rule of thumb is to add 10 entries for each downstream node.
CTDRV=	Defines the name of the communications driver used to communicate with the channel-attached PDM node. Set to DMECTCTC for nodes with integrated channel adapters, or to DMECTCTS for nodes using InfoSpeed Data Gateway connections.
OS=	Defines the operating system type of the node on the other side of this connection. Allowable values are UNIX and NT. The default is UNIX.
UNIT=	Defines the starting device number used for communications. This device number and the next in sequence will be used by the communications task. We recommend using the lowest device numbers for the communication tasks.
RAS=	Specifies whether a remote access server is to be defined for this communications path. The default value is N (no). If you are using any of the UNIX or NT clients or node FTP server through this communications path, then specify as RAS=Y.

Table 7. DMETABLE SDEF Statement Parameter Names and Explanation

SDEF Statement	
With label	**Defines**
SEQTEXT	Default values for text files (ASCII).
SEQBIN	Default values for nontext files (BINARY).
PDSTEXT	Default values for text files (ASCII) when a new PDS or PDSE is being created.
PDSBIN	Default values for nontext files (BINARY) when a new PDS or PDSE is being created.
TAPETEXT	Default values for text files (ASCII) with LSITE TAPE specified.
TAPEBIN	Default values for nontext files (BINARY) with LSITE TAPE specified.
SYSTEM	Default values if any one or all of the above are not coded. (Not used if all of the above are coded.)
Parameter	**Explanation**
ATYPE=	Specifies allocation type. Allowable values are cylinder, track, block, and tape. Use tape only on the TAPEBIN and TAPETEXT labels.
BLKSIZE=	Specifies the default block size of the MVS data set.
DIRBLKS=	Specifies the default number of directory blocks for new partitioned data sets. This keyword should be used only on the PDSBIN and PDSTEXT statements.
LRECL=	Specifies the default record length of the MVS data set.
PRIMARY=	Specifies the default primary allocation in the units specified by the ATYPE parameter if ATYPE is not tape.
SECONDARY=	Specifies the default secondary allocation in the units specified by the ATYPE parameter if ATYPE is not tape.
RECFM=	Specifies the default record format of the MVS data set. Allowable values are F, FB, FA, FBA, V, VB, VBA and U.
UNIT=	Specifies the default unit name.
VOLUME=	Specifies the default volume serial number.

The following are samples of each type of statement listed in Table 6 on page 52 and in Table 7:

```
          smf   stor=129,retr=130,ren=131,dele=132
          maxt  18
          priv  (ibmuser1,ibmuser2,ibmuser3)
 pisces   hdef  iodrv=dmeioctc,comm=com1,ftp=y,unit=(902,6),
                subhost=gemini
 gemini   hdef  iodrv=dmeiocts,comm=com2,ftp=y,unit=(90a,6),
                subhost=pisces
 taurus   hdef  iodrv=dmeiocts,comm=com3,ftp=y,unit=(912,6),os=nt
 com1     cdef  type=ctc,ctdrv=dmectctc,ras=y,qsize=200,unit=900
 com2     cdef  type=ctc,ctdrv=dmectcts,ras=y,qsize=200,unit=908
 com3     cdef  type=ctc,ctdrv=dmectcts,ras=y,qsize=200,unit=910
 seqtext  sdef  atype=track,primary=1,secondry=1,recfm=fb,lrecl=80,
                blksize=8000,unit=sysda
 pdstext  sdef  atype=track,primary=1,secondry=1,recfm=fb,lrecl=80,
                blksize=8000,dirblks=10,unit=sysda
 seqbin   sdef  atype=track,primary=2,secondry=1,recfm=vb,
                blksize=23008,lrecl=23004,unit=sysda
 pdsbin   sdef  atype=track,primary=2,secondry=1,recfm=vb,
                blksize=23008,lrecl=23004,unit=sysda,dirblks=10
tapetext sdef  atype=tape,unit=tape,lrecl=32756,blksize=32760,
                recfm=vb
tapebin  sdef  atype=tape,unit=tape,lrecl=32756,blksize=32760,
                recfm=vb
system   sdef  atype=track,primary=2,secondry=1,recfm=vb,
                blksize=23008,lrecl=23004,unit=sysda
```

Notes:

1. If you want PDM to write SMF records, code one, and only one, SMF statement. Allowable record types are 129-255 (IBM has reserved 1-128). A record type of zero (0) specifies that no SMF records are written for this function, or to turn off SMF accounting, the SMF statement can be omitted.
2. One PRIV statement can be optionally coded. Users can cancel their own transactions while privileged users can cancel any request, cancel from the ISPF Event Manager Panel, and change the state of MVS Transfer Paths and Communications Access Services Tasks.
3. One MAXT statement can optionally be coded. After PDM is started, the modify operator command can be used to change the number of tasks dynamically.
4. HDEF statements define all of the PDM nodes. There is one HDEF statement that defines each channel-attached node. It is recommended that the node name match the host name of the node. The name can be up to 16 characters long and is case insensitive.
5. Multiple UNIT= keywords may be coded on HDEF statements in the case where nonsequential device numbers are desired.
6. The CDEF statement defines the MVS communications task. One communications task can service many MVS transfer paths but a transfer path can use only one communications task. The MVS communications task uses two subchannels for reading and writing messages to that node defined on the CDEF statement.
7. The label on the CDEF statement must match the value of the COMM= parameter on the associated HDEF statement.
8. PDM selects one SDEF statement whenever a TCP/IP FTP client needs to create a new MVS data set and the user has not specified site parameters.
9. The SDEF statement is necessary only if using the node FTP server.
10. For parameters omitted from SITE and SDEF, SMS can provide defaults according to a data class or storage class. Refer to *DFSMS/MVS DFSMSdfp Storage Administration Reference*, SC26-4920, for more information.

6.3.2 HDETABLE Initialization Sample

Figure 26 on page 56 is a sample of a configuration of five remote nodes connected to the MVS server. The member DMETTWO from SDMESAMP was copied and modified to fit this installation.

```
//DMETFIVE JOB ,'PDMSYS',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=H,CLASS=A
//**********************************************************************
//*       INITIALIZE THE HOST DEFINITION TABLE                         *
//**********************************************************************
//TABLE    EXEC PGM=DMETABLE,REGION=1024K
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//DMETABLE DD  DSN=PDM.HDETABLE,DISP=SHR
//SYSIN DD  *
************************************************************************
*            PDM HOST TABLE CREATE                                     *
************************************************************************
*                                                                      *
*   For a detailed explanation of the keywords and values, refer to    *
*   the PDM Installation and Customization Guide.                      *
*                                                                      *
************************************************************************
*   Define SMF record types. To disable SMF accounting, comment out    *
*   the 'smf' statement. IBM has reserved values below 129.            *
*                                                                      *
*        SMF   STOR=129,RETR=130,REN=131,DELE=132
*                                                                      *
************************************************************************
************************************************************************
*   Define maximum number of concurrent transfer tasks - 'maxt' stmt  *
*   You may code 255, which will cause as many tasks as there are     *
*   paths to be started; however, these tasks take below the line     *
*   storage - a 4 MB region will run approximately 26.                *
*                                                                      *
         maxt  20
*                                                                      *
************************************************************************
*   Define define privileged MVS user IDs on the 'priv' statement.    *
*                                                                      *
         priv  (ibmuser,user1,user2)
*                                                                      *
************************************************************************
```

Figure 26 (Part 1 of 2). Sample JCL to Configure the HDETABLE.

```
*   Define the channel-attached nodes on the ′hdef′ statement.        *
*                                                                     *
* Note: All fields will be converted to uppercase with the exception  *
*   of the name of the hdef statement and the values for the subhost  *
*   keyword.                                                          *
pisces   hdef  iodrv=dmeioctc,comm=com1,ftp=y,unit=(902,6),
               subhost=(gemini,kingpin,hermes)
gemini   hdef  iodrv=dmeiocts,comm=com2,ftp=y,unit=(912,2),
               subhost=(pisces,kingpin,hermes)
hermes   hdef  iodrv=dmeiocts,comm=com3,ftp=y,unit=(916,2)
kingpin  hdef  iodrv=dmeiocts,comm=com4,ftp=y,unit=(91a,2)
d06d8lm35 hdef iodrv=dmeiocts,comm=com5,ftp=y,unit=(91e,2),os=nt
*                                                                     *
***********************************************************************
*   Define MVS communications tasks and remote access servers         *
*   on the ′cdef′ statement.                                          *
*   In column one is the name used in the matching hdef statement     *
*   ′comm=′ keyword.                                                  *
com1     cdef  type=ctc,ctdrv=dmectctc,ras=y,qsize=200,unit=900
com2     cdef  type=ctc,ctdrv=dmectcts,ras=y,qsize=200,unit=910
com3     cdef  type=ctc,ctdrv=dmectcts,ras=y,qsize=200,unit=914
com4     cdef  type=ctc,ctdrv=dmectcts,ras=y,qsize=200,unit=918
com5     cdef  type=ctc,ctdrv=dmectcts,ras=y,qsize=200,unit=91c,os=nt
*                                                                     *
***********************************************************************
*   Define default site parameters for the FTP server if desired.     *
*   These are used to eliminate the need for FTP client users to know *
*   about specific MVS dataset allocation values.                     *
*                                                                     *
*   To disable the defaults, either comment-out these statements or   *
*   delete them.                                                      *
*                                                                     *
seqtext  sdef  atype=track,primary=1,secondry=1,recfm=fb,lrecl=80,
               blksize=8000,unit=sysda
pdstext  sdef  atype=track,primary=1,secondry=1,recfm=fb,lrecl=80,
               blksize=8000,dirblks=10,unit=sysda
seqbin   sdef  atype=track,primary=2,secondry=1,recfm=vb,
               blksize=23008,lrecl=23004,unit=sysda
pdsbin   sdef  atype=track,primary=2,secondry=1,recfm=vb,
               blksize=23008,lrecl=23004,unit=sysda,dirblks=10
tapetext sdef  atype=tape,unit=tape,lrecl=32756,blksize=32760,
               recfm=vb
tapebin  sdef  atype=tape,unit=tape,lrecl=32756,blksize=32760,
               recfm=vb
system   sdef  atype=track,primary=2,secondry=1,recfm=vb,
               blksize=23008,lrecl=23004,unit=sysda
/*
//
```

Figure 26 (Part 2 of 2). Sample JCL to Configure the HDETABLE.

This sample produces the following report:

```
DMET006I HOST TABLE HAS BEEN SUCCESSFULLY BUILT
DMET008I THE FOLLOWING MAPS HAVE BEEN GENERATED:

    PATH             ID DVC-ADDR CDEF     SUBHOSTS
    pisces           00          COM1     gemini
                                          kingpin
                                          hermes
    pisces           01   0902   COM1     gemini
                                          kingpin
                                          hermes
    pisces           02   0903   COM1     gemini
                                          kingpin
                                          hermes
    pisces           03   0904   COM1     gemini
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    pisces           04   0905   COM1     gemini
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    pisces           05   0906   COM1     gemini
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    pisces           06   0907   COM1     gemini
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    gemini           00          COM2     pisces
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    gemini           01   0912   COM2     pisces
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    gemini           02   0913   COM2     pisces
                                          nome
                                          lutetium
                                          kingpin
                                          hermes
    hermes           00          COM3
    hermes           01   0916   COM3
    hermes           02   0917   COM3
    kingpin          00          COM4
    kingpin          01   091A   COM4
    kingpin          02   091B   COM4
    d06d8lm35        00          COM5
    d06d8lm35        01   091E   COM5
    d06d8lm35        02   091F   COM5
```

```
COMMTASK TYPE RAS   INDEV   OUTDEV  READAPPL WRTEAPPL      PATH
COM1     CTC   Y    0900     0901                     pisces
                                                      pisces
                                                      pisces
                                                      pisces
                                                      pisces
                                                      pisces
                                                      pisces
COM2     CTC   Y    0910     0911                     gemini
                                                      gemini
                                                      gemini
COM3     CTC   Y    0914     0915                     hermes
                                                      hermes
                                                      hermes
COM4     CTC   Y    0918     0919                     kingpin
                                                      kingpin
                                                      kingpin
COM5     CTC   Y    091C     091D                     d06d8lm35
                                                      d06d8lm35
                                                      d06d8lm35
```

The JCL in Figure 26 on page 56 defines the configuration in Figure 27.

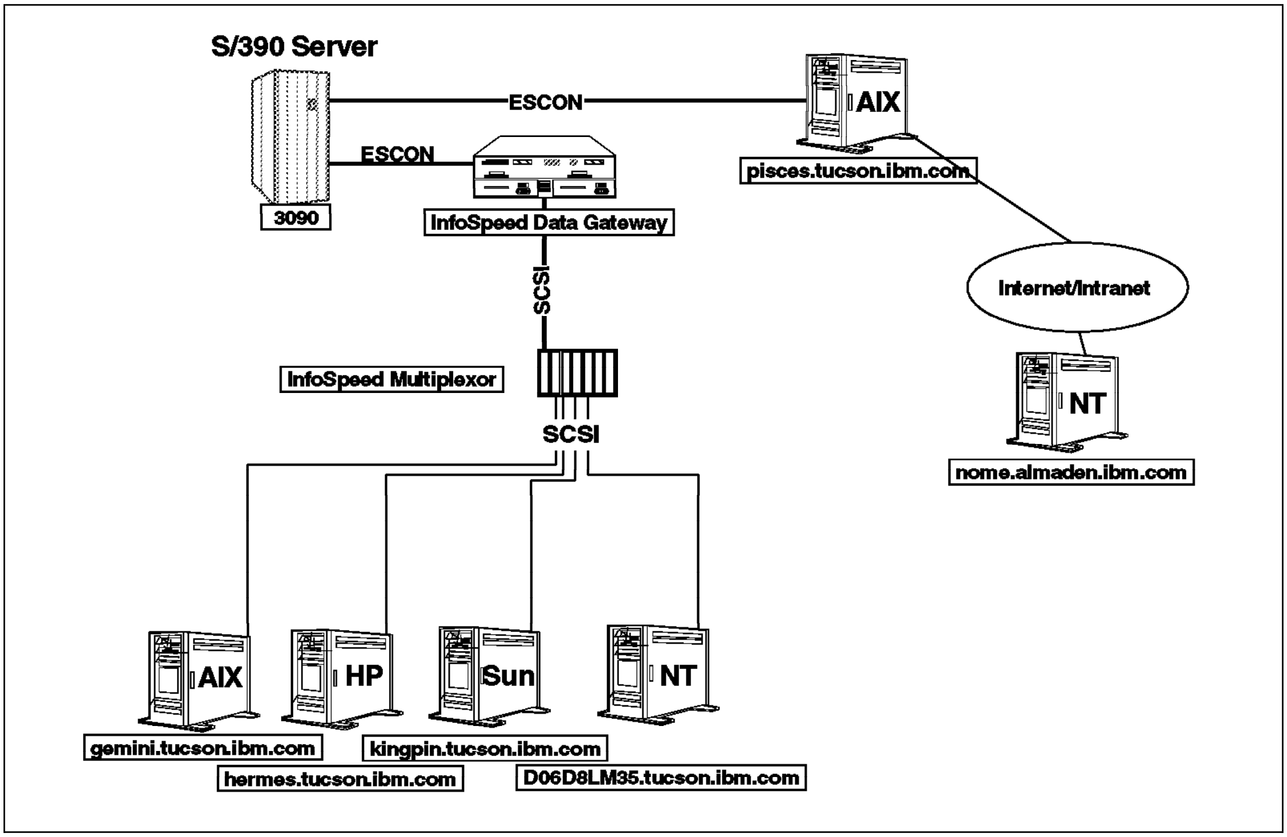


Figure 27. Sample MVS PDM Configuration

6.4 PDM MVS Subsystem Installation Checklist

Before attempting to bring up the system to an active state, review the following checklist:

__ 1. Verify prerequisites.

__ 2. IOCP/HCD definition entered.

__ 3. Define DMES as a subsytem to MVS.

__ 4. Disable the missing interrupt handler for all devices.

__ 5. Load MVS libraries.

__ 6. Link MVS libraries.

__ 7. SDMELINK in authorized library list?

__ 8. Allocate and initialize QUEUE.

__ 9. Allocate and initialize HDETABLE.

__ 10. Customize start-PDM JCL.

__ 11. MVS data set security in place?

__ 12. Install and customize ISPF screens and libraries.

__ 13. Edit and build translate tables.

__ 14. SMF considerations accounted for?

__ 15. Install the REXX interface.

__ 16. Initialize and configure HDETABLE.

6.5 PDM MVS Subsystem Operation

Once the PDM MVS software has been installed and the HDETABLE configured, the PDM subsystem can be started. First, verify that the subsystem has been added to the system. Enter the following command:

```
D SSI,SUBSYS=DMES
```

The output from the display command should look like the following:

```
SUBSYS=DMES
   DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=ACCEPT
```

If the DMES subsystem does not exist, then either DMES was not defined to the system or the subsystem was not dynamically added. Refer to 6.2.3, “Define PDM as a Subsystem” on page 45. At this point, DMES can be added dynamically with the following operator command:

```
SETSSI ADD,S=DMES
```

The subsystem must be defined as dynamic to accept SETSSI commands. The initial SETSSI ADD command adds DMES as a dynamic subsystem as well as the SUBSYS SUBNAME (DMES) entry in the IEFSSNxx member in SYS1.PARMLIB. If the older positional parameter form of the IEFSSNxx member is used, the subsystem is not defined as dynamic and no SETSSI commands are accepted.

6.5.1 Starting and Stopping the MVS DMES Subsystem

After the subsystem has been added, start the PDM address space, by entering the following command, where *DMESYS* is the name of the started procedure:

```
S DMESYS
```

During startup, the subsystem issues a number of operator messages that depict the status of various steps in the initialization process. Startup is complete when the following message is issued:

```
 DME001I SUPERVISOR INITIALIZATION COMPLETE
```

An unsuccessful startup of the address space may be the result of any of the following:

- Incorrect PDM data set names are coded in the DMESYS procedure.
- The SDMELINK library is not an authorized library.
- The MVS ID associated with the DMESYS address space is not authorized for the PDM data sets.
- The DMES subsystem was not defined to the system.

Refer to *MVS System Messages*, GC28-1784 through GC28-1788, for abnormal-termination (abend) messages and corrective actions.

To stop the PDM subsystem, issue the standard MVS stop command using the name of the started task. Avoid using the cancel command unless absolutely necessary. Enter the following command:

```
P DMESYS
```

PDM acknowledges the stop command with the following message:

```
 DME007I SHUTDOWN REQUEST IS ACKNOWLEDGED
```

The modify command can be used to change the number of concurrent transfer operations PDM can perform. It overrides the MAXT setting in the HDETABLE data set. The following is the format of the command, where *DMESYS* is the name of the started procedure and *nn* is the new number of the concurrent transfers:.

```
MODIFY DMESYS,TASKS=nn
```

When PDM has reconfigured the tasks, you receive the following messages:

```
 DMESYS   DME015I THERE ARE nnn ACTIVE TRANSFER TASKS
 DMESYS   DME006I MEMORY IS AVAILABLE FOR AN ADDITIONAL iii TRANSFER TASKS
```

In these messages, *nnn* is the total number of tasks that are started and *iii* is the approximate number of additional tasks that could be started (given the current specification for region size).

Note: You cannot set the number of concurrent transfers to be greater than the total number of defined connections. If you do, PDM uses the number of defined connections. Therefore, if you want to specify the maximum, use 99 as the number of tasks.

6.5.2 Verifying MVS and Remote Node Connectivity

Upon startup of the DMES address space, PDM initializes the communications tasks and then tries to establish contact with the PDM remote nodes. Once contact has been established, transfer tasks are allocated to the transfer paths; the MVS system log and PDM log indicate whether the connection is successful. The PDM software must be started on the remote node before a communication link can be established between systems. Pay close attention to the MVS system console log because messages are issued to indicate the status of the startup. The PDM ISPF interface can be used to monitor the state of all communications and transfer paths (refer to 10.1.2, “Using the MVS ISPF Client (DMESISPF)” on page 111). When the communications link is established, the following message is displayed:

```
DME024I COMMUNICATIONS TASK nnnnnnnn HAS ESTABLISHED CONTACT
AND IS NOW ACTIVE
```

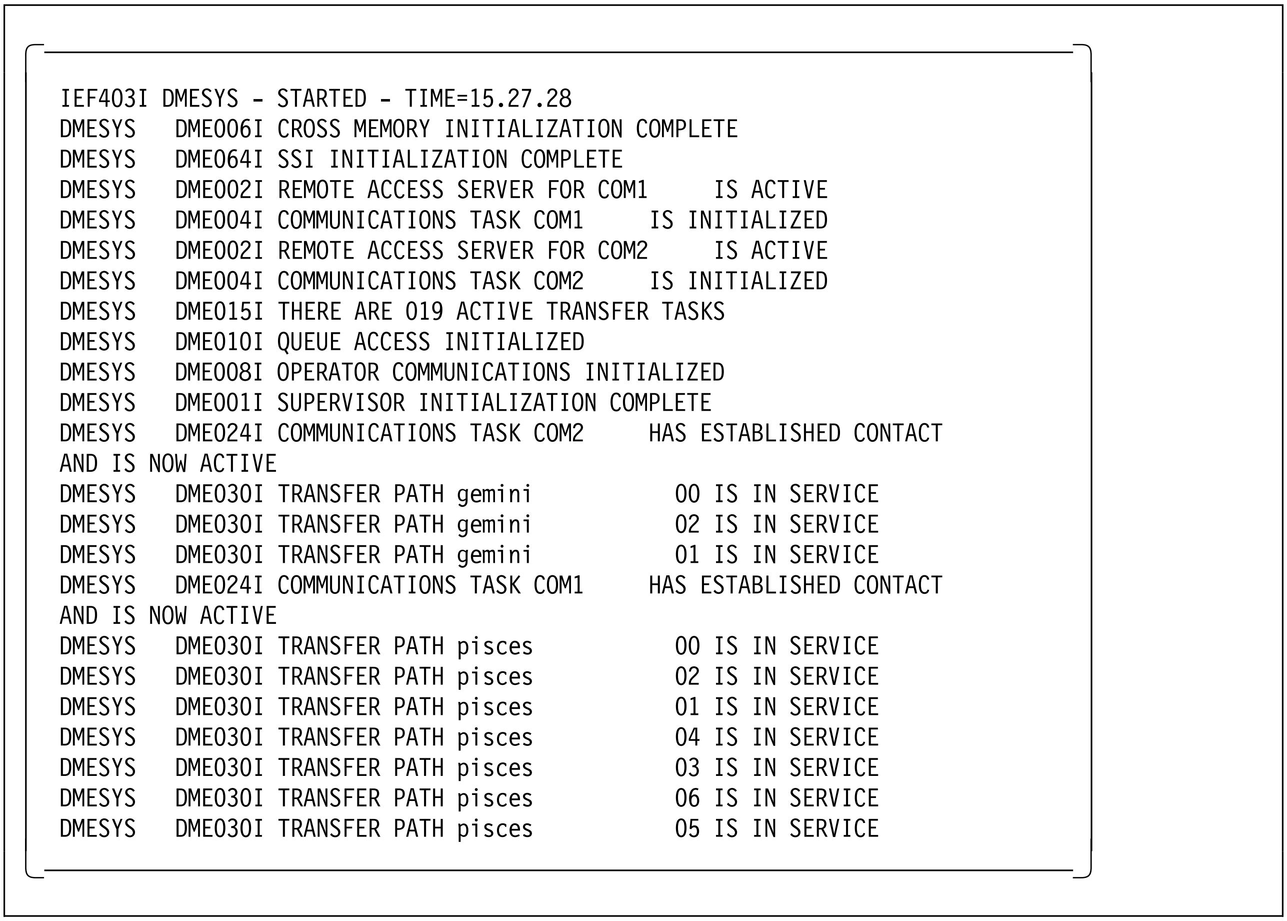
where nnnnnnn is the name of the communications task (the label that is on the CDEF statement for this node).

When communications is established, a CONNECT function request is sent with a request ID of 00000000. Until a connect request is successfully completed, a transfer path is not marked online and is not available for use. This feature verifies that channel addresses are properly configured and functional. For each transfer path, the following message is displayed:

```
DME030I TRANSFER PATH nnnnnnnnnnnnnnnn ii IS IN SERVICE
```

where nnnn... is the name of the transfer path (the label that is on the HDEF statement for this node) and ii is the path number assigned by the HDETABLE initiation program (see the SYSOUT output).

The messages in Figure 28 on page 63 indicate a successful startup of the DMESYS address space and successful connection of the MVS PDM subsystem to two PDM remote nodes. The messages in the example are logged in the MVS system log and the PDM log.



```
IEF403I DMESYS - STARTED - TIME=15.27.28
DMESYS   DME006I CROSS MEMORY INITIALIZATION COMPLETE
DMESYS   DME064I SSI INITIALIZATION COMPLETE
DMESYS   DME002I REMOTE ACCESS SERVER FOR COM1     IS ACTIVE
DMESYS   DME004I COMMUNICATIONS TASK COM1     IS INITIALIZED
DMESYS   DME002I REMOTE ACCESS SERVER FOR COM2     IS ACTIVE
DMESYS   DME004I COMMUNICATIONS TASK COM2     IS INITIALIZED
DMESYS   DME015I THERE ARE 019 ACTIVE TRANSFER TASKS
DMESYS   DME010I QUEUE ACCESS INITIALIZED
DMESYS   DME008I OPERATOR COMMUNICATIONS INITIALIZED
DMESYS   DME001I SUPERVISOR INITIALIZATION COMPLETE
DMESYS   DME024I COMMUNICATIONS TASK COM2     HAS ESTABLISHED CONTACT
AND IS NOW ACTIVE
DMESYS   DME030I TRANSFER PATH gemini           00 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH gemini           02 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH gemini           01 IS IN SERVICE
DMESYS   DME024I COMMUNICATIONS TASK COM1     HAS ESTABLISHED CONTACT
AND IS NOW ACTIVE
DMESYS   DME030I TRANSFER PATH pisces           00 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH pisces           02 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH pisces           01 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH pisces           04 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH pisces           03 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH pisces           06 IS IN SERVICE
DMESYS   DME030I TRANSFER PATH pisces           05 IS IN SERVICE
```

Figure 28. PDM LOG Messages Indicating a Successful DMESYS Startup and Connection. The remote nodes are pisces and gemini with 19 transfer tasks defined. COM1 is the Communications Task associated with pisces and COM2 is the Communications Task associated with gemini.

Some of the reasons why the DMES subsystem may fail to connect to a remote node are:

- The HDETABLE was not properly configured. Verify the output listing from the JCL submitted to ensure the job completed successfully. Also verify that all values for all parameters are correct, including UNIT numbers and COMM labels.
- The channels are not online to MVS. Display the devices on MVS and vary online (if offline to MVS).
- The physical connections between MVS and the remote nodes are not properly attached. Check all ESCON and SCSI connections between MVS and the remote node. If using a IBM InfoSpeed Data Gateway or IBM InfoSpeed Multiplexor, verify that the connections are in the correct slots and are properly attached, and that any in-line SCSI terminators have been installed.
- The HCD was incorrectly defined. Verify that the device addresses are correct, are correctly configured, and are properly defined.
- The IBM InfoSpeed Data Gateway (if used) was incorrectly configured. There is a mapping in the gateway hardware between the channel address and the SCSI address or LUN. The channel addresses must be properly mapped to the connected node.

- The PDM software has not been started on the remote node. Start PDM on the remote node.
- The channel addresses are inactive or unavailable from the connected node′s point of view. With an IBM InfoSpeed Data Gateway connection, use the UNIX *mt* command to issue a rewind command to the logical tape device. If there is no response from the operating system, the device is unavailable.

Chapter 7. Install and Configure PDM Channel-Attached Nodes

This chapter describes the installation and configuration of the IBM exPress Data Mover software on all the UNIX or Windows NT platforms which support it: AIX, HP, Sun and Windows NT.

7.1 AIX Installation Planning

There are four main aspects to planning an AIX installation.

7.1.1 Install the IBM exPress Data Mover

IBM exPress Data Mover is shipped in tar compressed format on either floppy disk or tape. To install the product, perform the following steps :

1. Log in as root.
2. Create the *pdmsys* user group and *pdmsys* user account:
   - Enter *smitty users* and add *pdmsys*.
   - Enter *smitty groups* and add *pdmsys*.
3. Run smit to install the PDM files:
   - Enter *smitty* or *smitty update_all*.
4. Select the proper device */dev/rmtx* or */dev/rfd0* as input device.
5. Select the default for all products.
6. When installation is finished, clock on OK as prompted on the screen.

7.1.2 Configure the ESCON Adapter

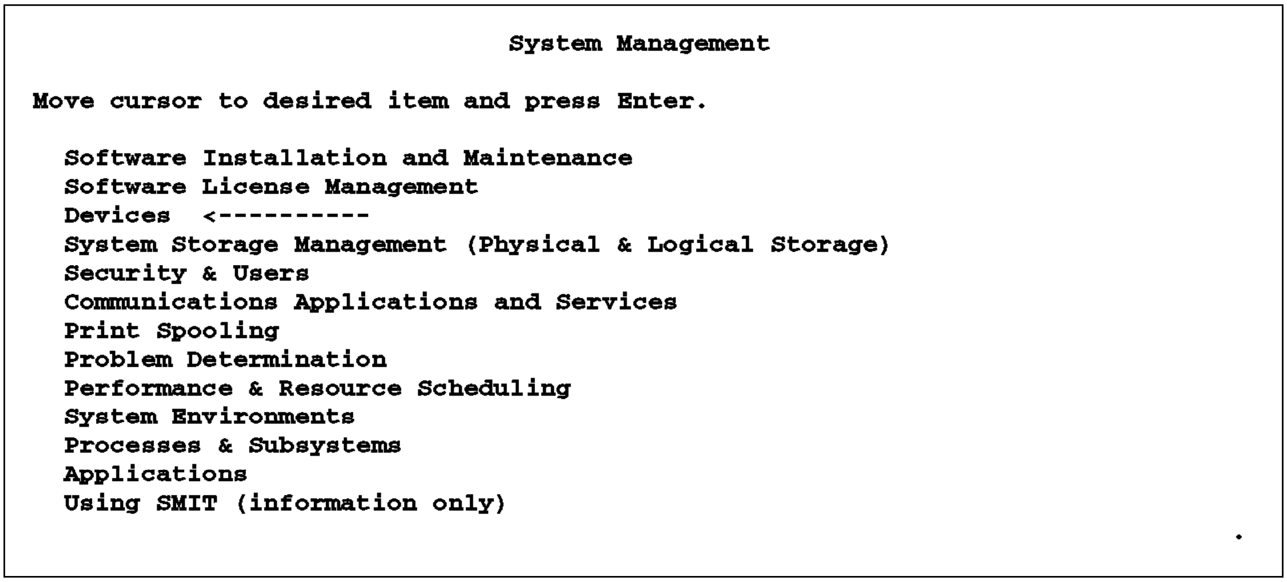
The communication paths over an ESCON channel are defined in AIX as a set of 3088 image/devices. The assigned adresses must match those used by the OS/390 administrator to define the ESCON subchannels. The address range defined in AIX must match the last two digits of the subchannel addresses that are defined on HCD in the S/390 system. For example, 520 to 527 need to be defined as 20 to 27 on the smit ESCON definitions. As for OS/390, PDM on AIX requires control paths and data paths, and there must be at least be as many data transfer paths as are defined in OS/390.

The first subchannel to be defined (at the lowest address) must match the UNIT=xxxx on the CDEF statement coded in the PDM host table on OS/390. The second subchannel (next in sequence) is also used for communication. The remaining paths are for data transfers, the first of which should match the UNIT=yyyy on the HDEF statement coded in the PDM host table on OS/390.

When adding the subchannels in AIX, each path has its own subchannel to be defined. Thus, if you plan to use four data transfer paths, you have to add six subchannels (two communication paths plus four data paths), using the smit screens as described in Figure 29 on page 66 through Figure 41 on page 71.

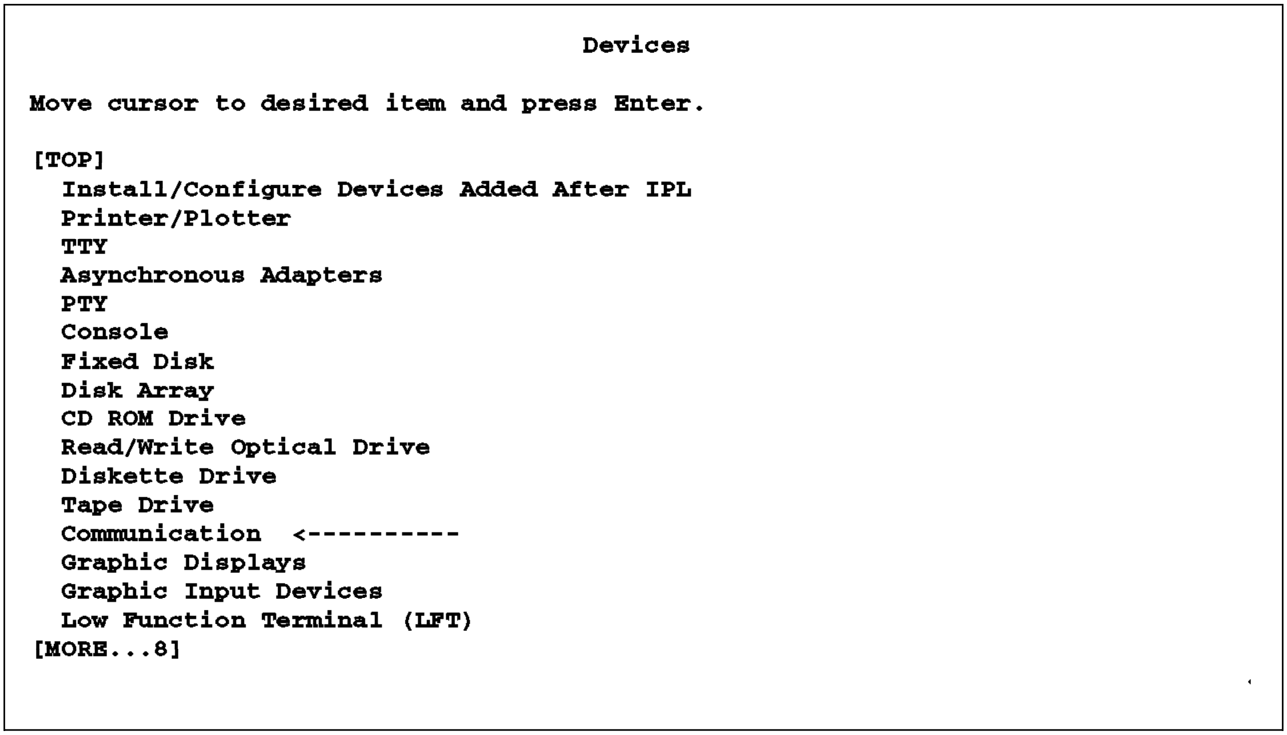
Then,

- Type *smitty* or *smitty esca* (to skip the two first screens)



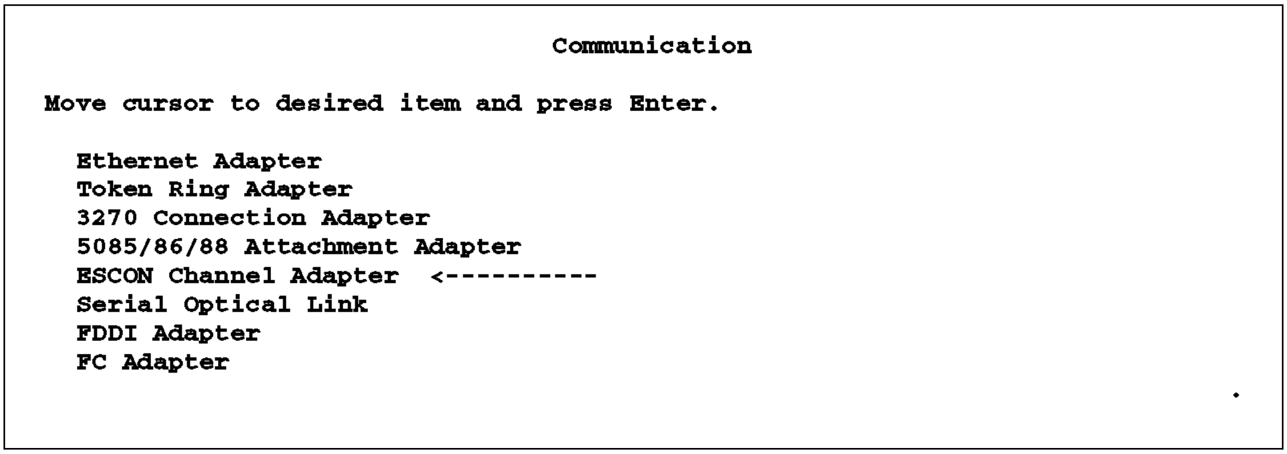
```
                                     System Management

 Move cursor to desired item and press Enter.

   Software Installation and Maintenance
   Software License Management
   Devices  <----------
   System Storage Management (Physical & Logical Storage)
   Security & Users
   Communications Applications and Services
   Print Spooling
   Problem Determination
   Performance & Resource Scheduling
   System Environments
   Processes & Subsystems
   Applications
   Using SMIT (information only)
```

Figure 29. SMIT System Management Screen



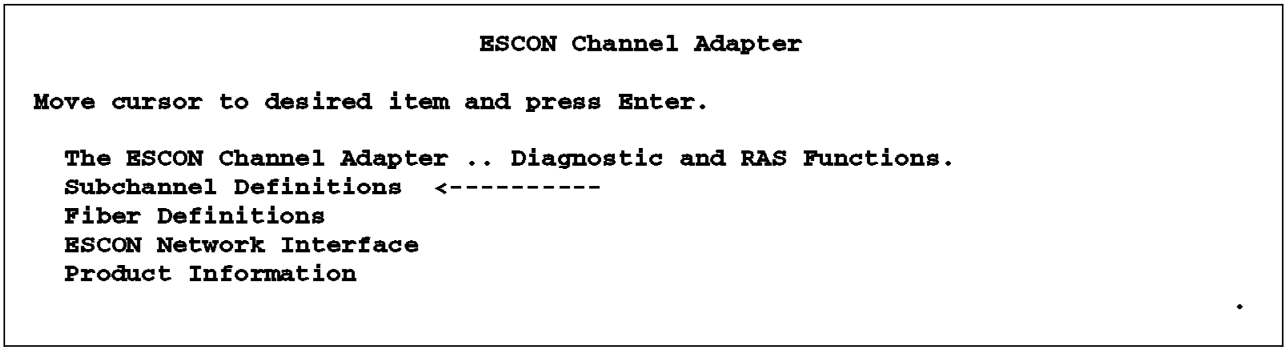
```
                                          Devices

 Move cursor to desired item and press Enter.

 [TOP]
   Install/Configure Devices Added After IPL
   Printer/Plotter
   TTY
   Asynchronous Adapters
   PTY
   Console
   Fixed Disk
   Disk Array
   CD ROM Drive
   Read/Write Optical Drive
   Diskette Drive
   Tape Drive
   Communication  <----------
   Graphic Displays
   Graphic Input Devices
   Low Function Terminal (LFT)
 [MORE...8]
```

Figure 30. SMIT Devices Screen



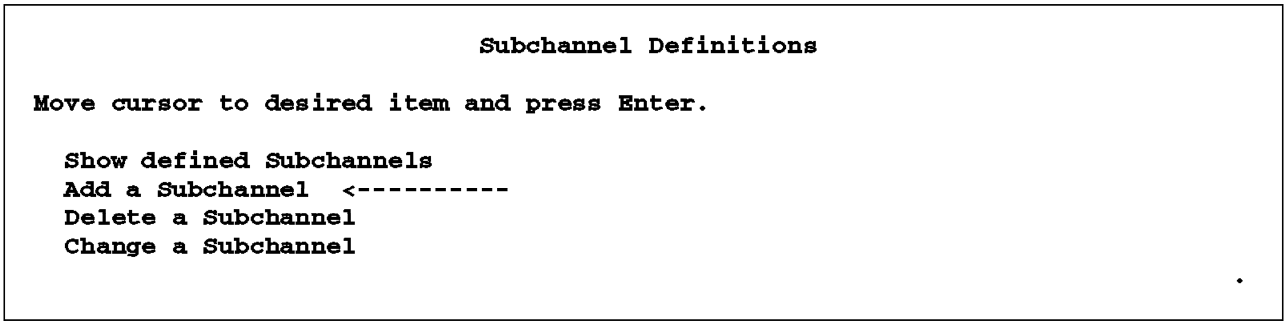
```
                         Communication

 Move cursor to desired item and press Enter.

   Ethernet Adapter
   Token Ring Adapter
   3270 Connection Adapter
   5085/86/88 Attachment Adapter
   ESCON Channel Adapter  <----------
   Serial Optical Link
   FDDI Adapter
   FC Adapter
```

Figure 31. SMIT Communication Screen



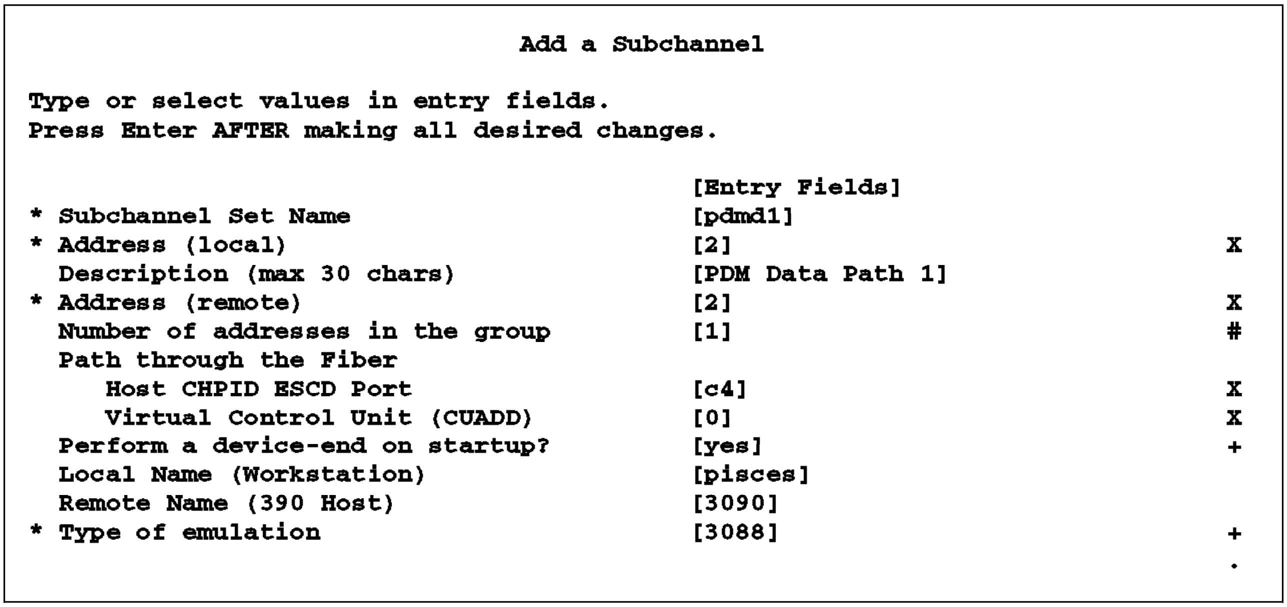
```
                      ESCON Channel Adapter

Move cursor to desired item and press Enter.

  The ESCON Channel Adapter .. Diagnostic and RAS Functions.
  Subchannel Definitions  <----------
  Fiber Definitions
  ESCON Network Interface
  Product Information
```

Figure 32. SMIT ESCON Channel Adapter Screen



```
                      Subchannel Definitions

Move cursor to desired item and press Enter.

  Show defined Subchannels
  Add a Subchannel  <----------
  Delete a Subchannel
  Change a Subchannel
```

Figure 33. SMIT Subchannel Definitions Screen

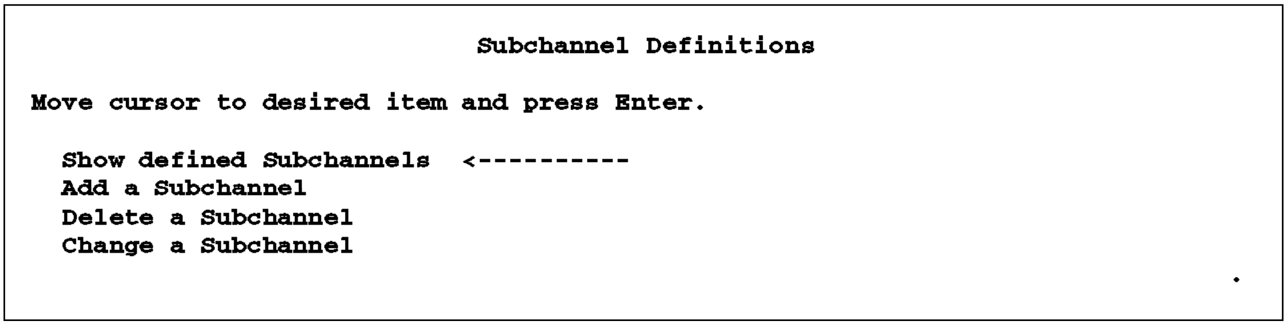


```
                          Add a Subchannel

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                           [Entry Fields]
* Subchannel Set Name                      [pdmd1]
* Address (local)                          [2]                        X
  Description (max 30 chars)               [PDM Data Path 1]
* Address (remote)                         [2]                        X
  Number of addresses in the group         [1]                        #
  Path through the Fiber
     Host CHPID ESCD Port                  [c4]                       X
     Virtual Control Unit (CUADD)          [0]                        X
  Perform a device-end on startup?         [yes]                      +
  Local Name (Workstation)                 [pisces]
  Remote Name (390 Host)                   [3090]
* Type of emulation                        [3088]                     +
                                                                      .
```

Figure 34. ESCON Subchannel Configuration Menus in SMIT

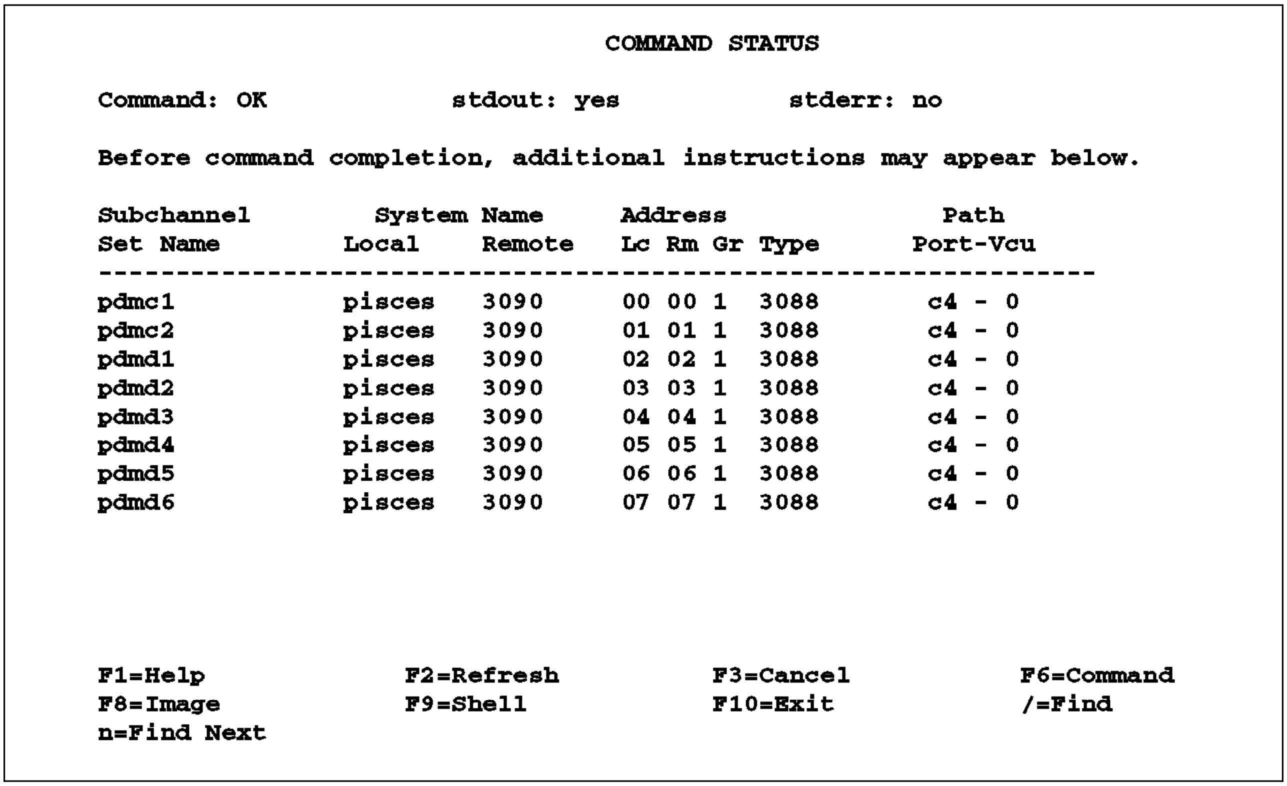
In Figure 34, we give an example of adding the third subchannel. The first two (communication subchannels) were arbitrarily named pdmc1 and pdmc2). As you can see, we use only one address (*number_of_addresses_in_the_group* parameter) per subchannel. The local address is the same as the remote address. The subchannel name is not set by PDM and should be named by the installer.

Once all subchannels are defined, select a display of them using the menu shown in Figure 35. The result should look like Figure 36 on page 69.



```
                       Subchannel Definitions

Move cursor to desired item and press Enter.

  Show defined Subchannels  <----------
  Add a Subchannel
  Delete a Subchannel
  Change a Subchannel
                                                                      .
```

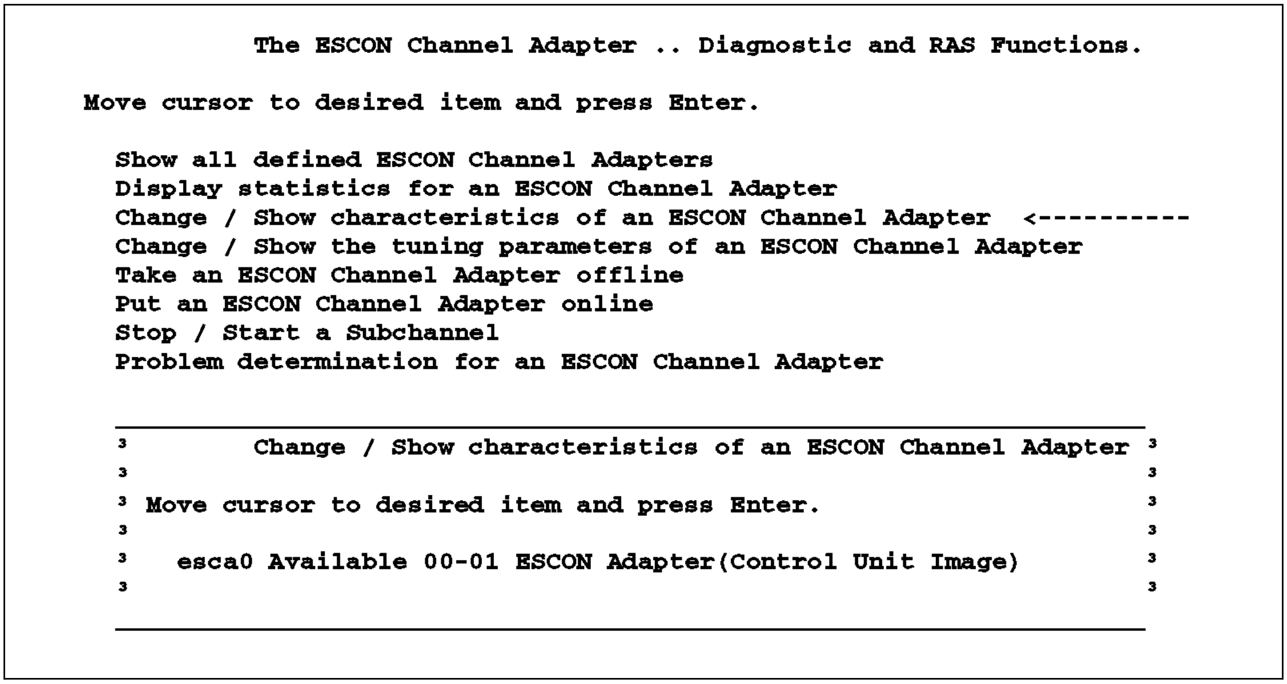
Figure 35. SMIT Subchannels Definitions



```
                              COMMAND STATUS

Command: OK            stdout: yes          stderr: no

Before command completion, additional instructions may appear below.

Subchannel        System Name     Address            Path
Set Name        Local    Remote    Lc Rm Gr Type      Port-Vcu
-----------------------------------------------------------------
pdmc1           pisces   3090      00 00 1  3088       c4 - 0
pdmc2           pisces   3090      01 01 1  3088       c4 - 0
pdmd1           pisces   3090      02 02 1  3088       c4 - 0
pdmd2           pisces   3090      03 03 1  3088       c4 - 0
pdmd3           pisces   3090      04 04 1  3088       c4 - 0
pdmd4           pisces   3090      05 05 1  3088       c4 - 0
pdmd5           pisces   3090      06 06 1  3088       c4 - 0
pdmd6           pisces   3090      07 07 1  3088       c4 - 0




F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
n=Find Next
```

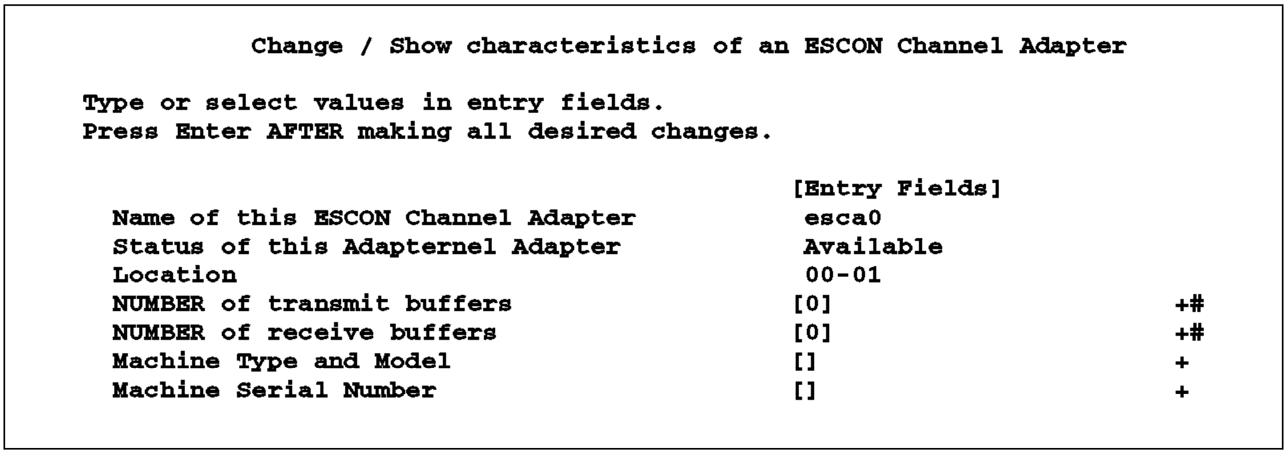
Figure 36. ESCON Subchannel Characteristics

Configure the CHARACTERISTICS of the ESCON adapter, using smit, as shown in Figure 37. The dialog is shown in Figure 38 on page 70. Set the Number of Transmit Buffers and the Number of Receive Buffers to zero. This allows the system to use the available buffers in the adapter.



```
           The ESCON Channel Adapter .. Diagnostic and RAS Functions.

Move cursor to desired item and press Enter.

  Show all defined ESCON Channel Adapters
  Display statistics for an ESCON Channel Adapter
  Change / Show characteristics of an ESCON Channel Adapter  <----------
  Change / Show the tuning parameters of an ESCON Channel Adapter
  Take an ESCON Channel Adapter offline
  Put an ESCON Channel Adapter online
  Stop / Start a Subchannel
  Problem determination for an ESCON Channel Adapter

  ___________________________________________________________________
  ³        Change / Show characteristics of an ESCON Channel Adapter ³
  ³                                                                  ³
  ³ Move cursor to desired item and press Enter.                     ³
  ³                                                                  ³
  ³   esca0 Available 00-01 ESCON Adapter(Control Unit Image)        ³
  ³                                                                  ³
  ___________________________________________________________________
```

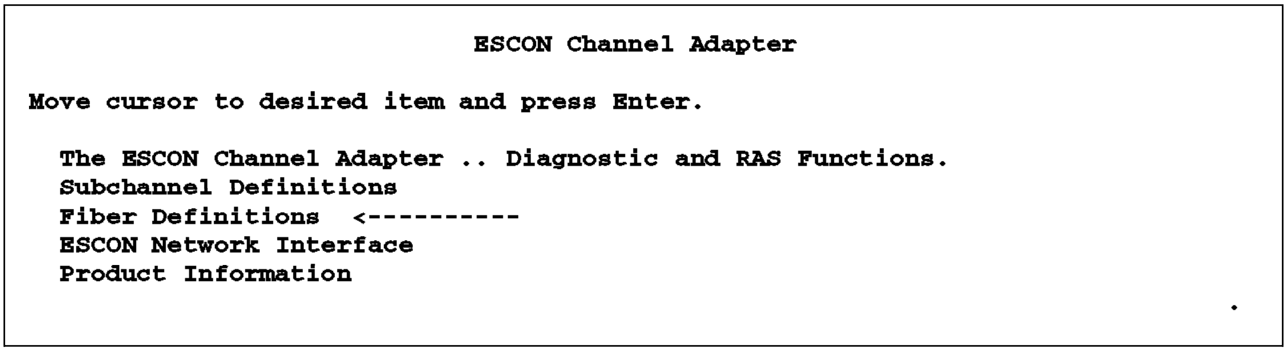
Figure 37. SMIT ESCON Channel Adapter Menu



```
            Change / Show characteristics of an ESCON Channel Adapter

    Type or select values in entry fields.
    Press Enter AFTER making all desired changes.

                                            [Entry Fields]
      Name of this ESCON Channel Adapter     esca0
      Status of this Adapternel Adapter      Available
      Location                               00-01
      NUMBER of transmit buffers            [0]                      +#
      NUMBER of receive buffers             [0]                      +#
      Machine Type and Model                []                       +
      Machine Serial Number                 []                       +
```

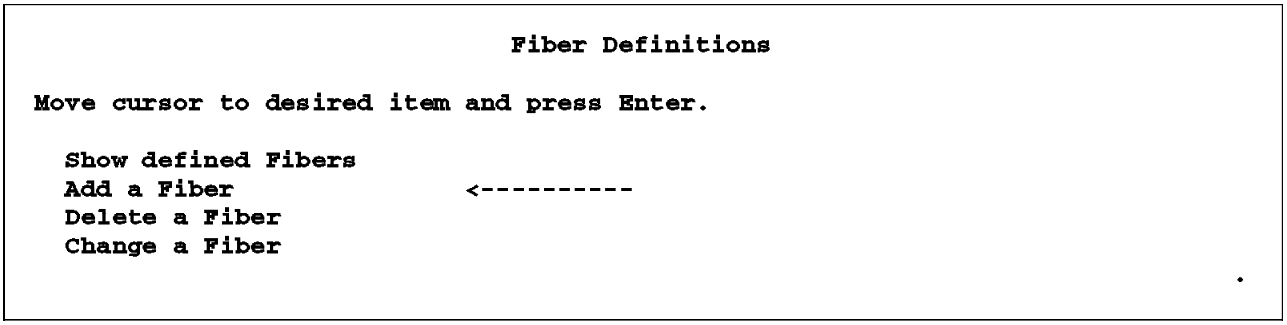
Figure 38. ESCON Subchannel Characteristics

Next, you need to add a single fiber definition that contains all of the subchannel names, using the menu shown in Figure 39. The dialog screens are shown in Figure 40 and in Figure 41 on page 71.



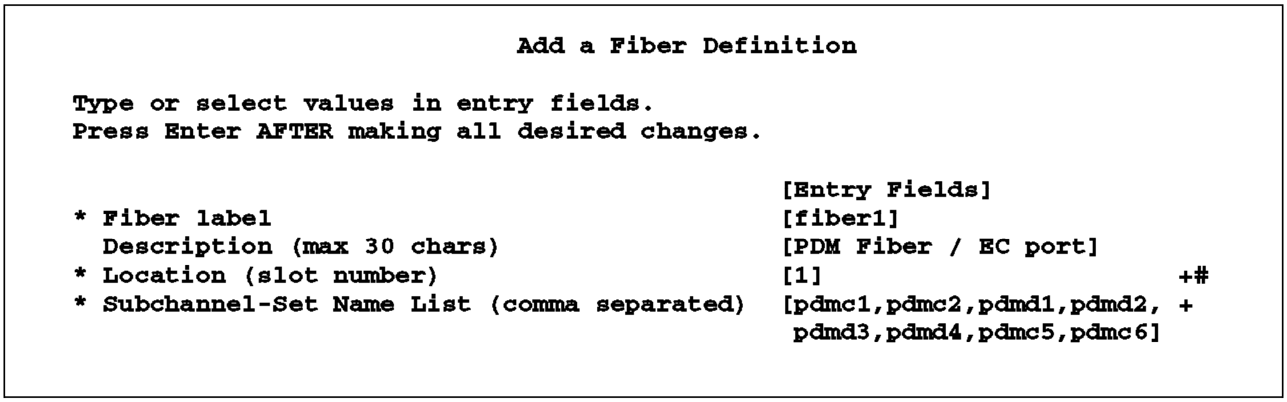
```
                         ESCON Channel Adapter

 Move cursor to desired item and press Enter.

   The ESCON Channel Adapter .. Diagnostic and RAS Functions.
   Subchannel Definitions
   Fiber Definitions  <----------
   ESCON Network Interface
   Product Information
                                                                  .
```

Figure 39. SMIT ESCON Channel Adapter Menu



```
                           Fiber Definitions

 Move cursor to desired item and press Enter.

   Show defined Fibers
   Add a Fiber            <----------
   Delete a Fiber
   Change a Fiber
                                                                  .
```

Figure 40. SMIT Fiber Definitions



```
                        Add a Fiber Definition

  Type or select values in entry fields.
  Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  * Fiber label                                   [fiber1]
    Description (max 30 chars)                    [PDM Fiber / EC port]
  * Location (slot number)                        [1]                    +#
  * Subchannel-Set Name List (comma separated)    [pdmc1,pdmc2,pdmd1,pdmd2, +
                                                   pdmd3,pdmd4,pdmc5,pdmc6]
```

Figure 41. Add a Fiber Dialog Screen

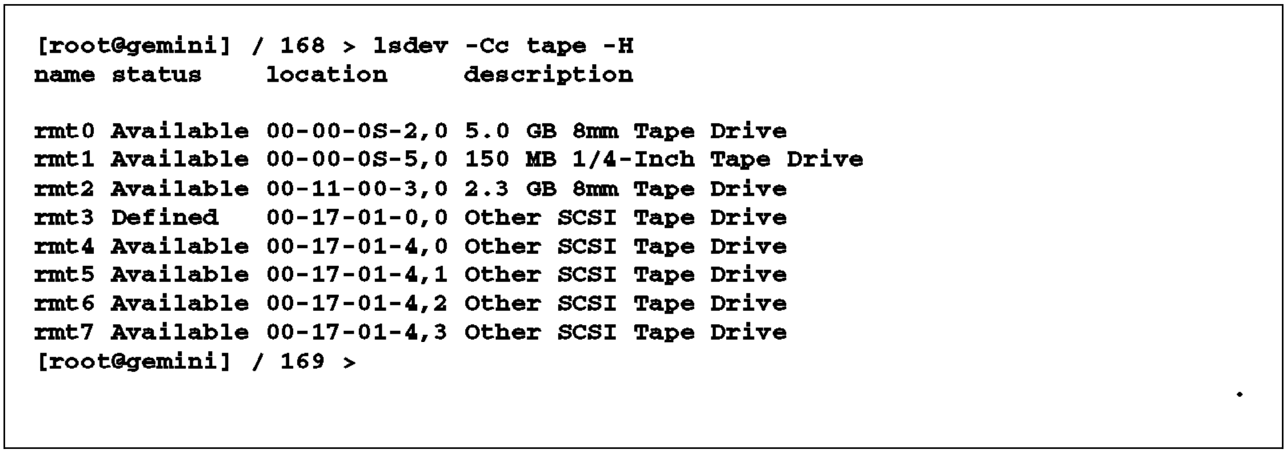
7.1.3 Configure the SCSI Adapter

The physical connection of the SCSI adapter either to the IBM InfoSpeed Data Gateway or to the IBM InfoSpeed Multiplexor should have been done according to Figure 20 on page 31 or Figure 21 on page 32. If not, do it now.

For AIX to detect the new SCSI devices that have been installed, you can either reboot or type cfgmgr on the command line. AIX creates the appropriate tape devices. The next steps are these:

1. List the devices that have been created:
   - Type *lsdev -Cc tape -H*

   This command should prompt something like Figure 42.



```
[root@gemini] / 168 > lsdev -Cc tape -H
name status    location     description

rmt0 Available 00-00-0S-2,0 5.0 GB 8mm Tape Drive
rmt1 Available 00-00-0S-5,0 150 MB 1/4-Inch Tape Drive
rmt2 Available 00-11-00-3,0 2.3 GB 8mm Tape Drive
rmt3 Defined   00-17-01-0,0 Other SCSI Tape Drive
rmt4 Available 00-17-01-4,0 Other SCSI Tape Drive
rmt5 Available 00-17-01-4,1 Other SCSI Tape Drive
rmt6 Available 00-17-01-4,2 Other SCSI Tape Drive
rmt7 Available 00-17-01-4,3 Other SCSI Tape Drive
[root@gemini] / 169 >
```

Figure 42. List of Tape Devices

Note: In Figure 42 the IBM InfoSpeed Data Gateway was added by AIX as rmt4-rmt7 as ″Other SCSI Tape Drive″.

2. Check the name of the new device (or devices) that have been created (rmtx) and modify it (or them) as follows:
   - Type *chdev -l rmtx -a block_size=0*
   - *Type chdev -l rmtx -a rwtimeout=900*
3. Verify your hardware configuration:
   - Type *mt -f /dev/rmtx rewind*

This command should be completed with no error.

7.1.4 Configure IBM exPress Data Mover

The first step is to install the software on the nodes that need it.

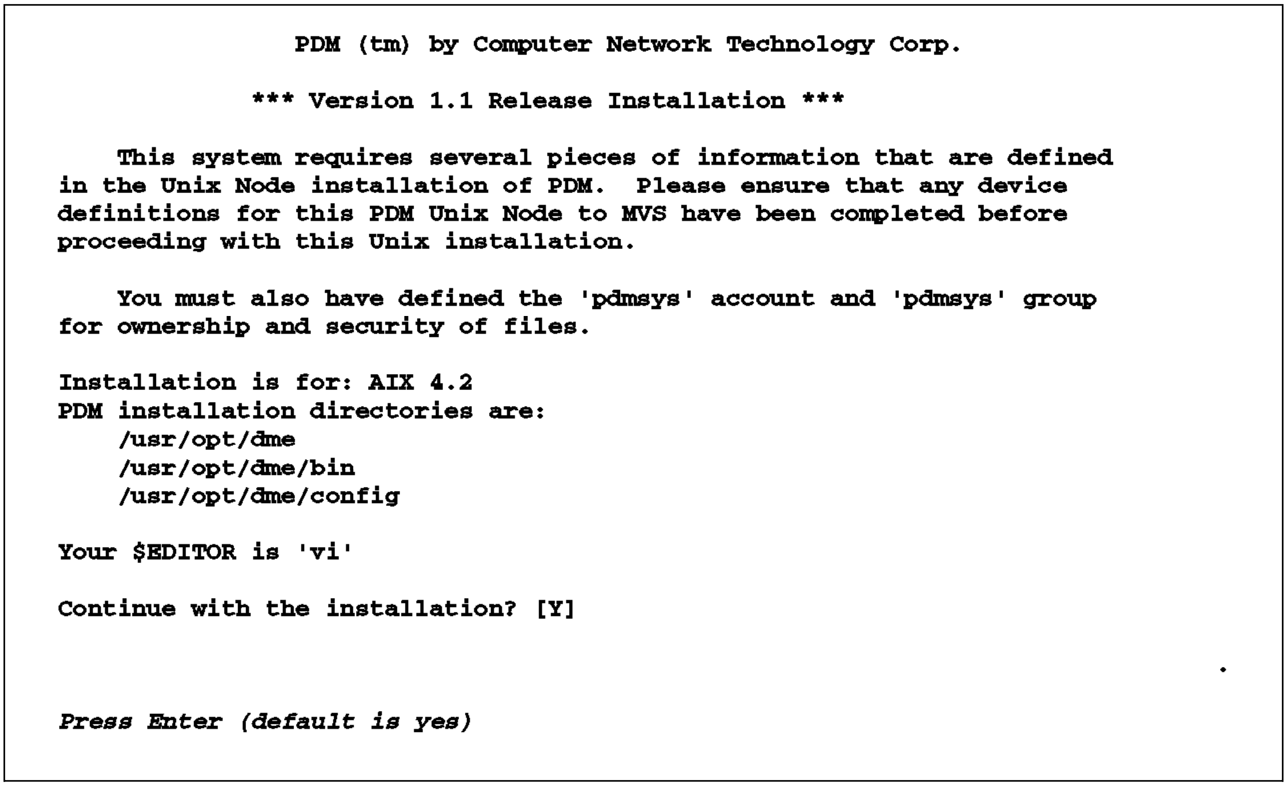
7.1.4.1 Run the Installation Program

Once the software is installed and the adapters have been correctly configured, you can run the installation program:

- Type *cd /usr/opt/dme*
- *Type ./install_pdm*

The *install_pdm* file is the main PDM installation program. It determines the system type, creates directories, sets the required authorities to the PDM daemons, edits the environment file during the installation process, and asks you to ″fill in the fields.″

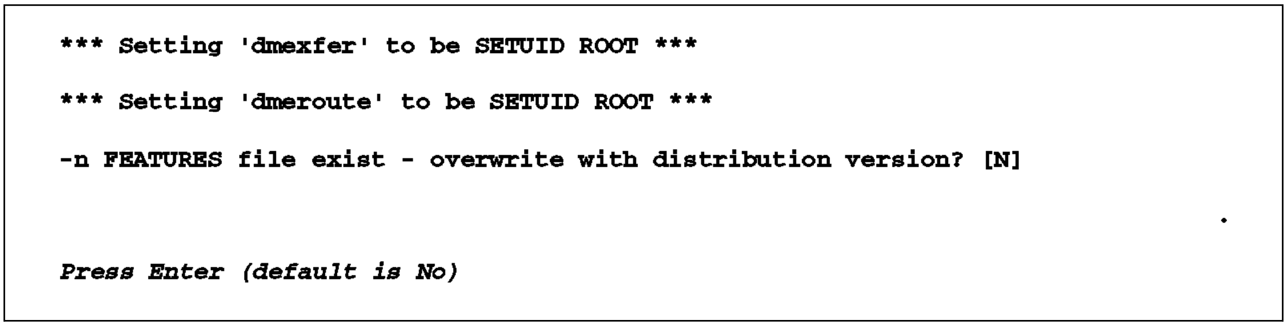
The first message to be prompted is shown in Figure 43.



```
                PDM (tm) by Computer Network Technology Corp.

             *** Version 1.1 Release Installation ***

    This system requires several pieces of information that are defined
in the Unix Node installation of PDM.  Please ensure that any device
definitions for this PDM Unix Node to MVS have been completed before
proceeding with this Unix installation.

    You must also have defined the 'pdmsys' account and 'pdmsys' group
for ownership and security of files.

Installation is for: AIX 4.2
PDM installation directories are:
    /usr/opt/dme
    /usr/opt/dme/bin
    /usr/opt/dme/config

Your $EDITOR is 'vi'

Continue with the installation? [Y]


Press Enter (default is yes)
```

Figure 43. PDM Installation Menu on AIX(1)

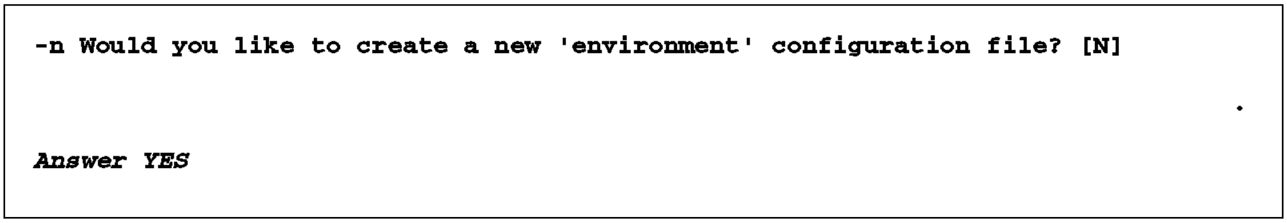
The next prompt is shown in Figure 44 on page 73.



```
*** Setting 'dmexfer' to be SETUID ROOT ***

*** Setting 'dmeroute' to be SETUID ROOT ***

-n FEATURES file exist - overwrite with distribution version? [N]


Press Enter (default is No)
```

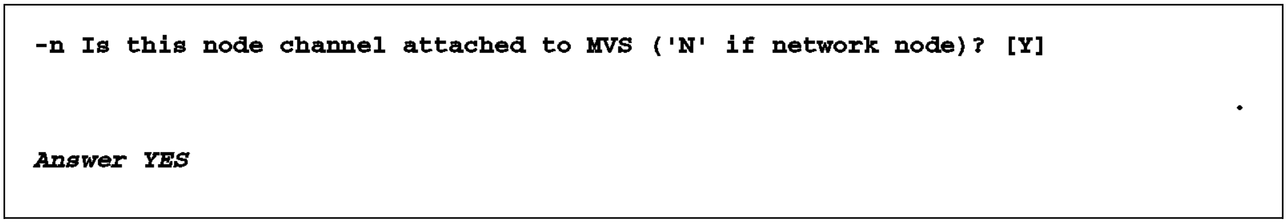
Figure 44. PDM Installation Menu on AIX(2)

Subsequent prompts are shown in Figure 45 and Figure 46.



```
-n Would you like to create a new 'environment' configuration file? [N]


Answer YES
```

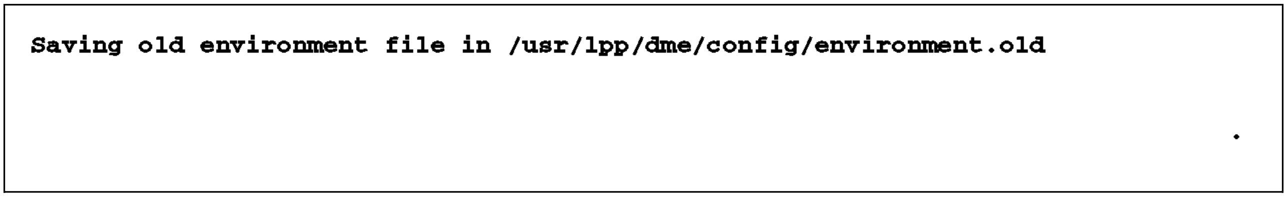
Figure 45. PDM Installation Menu on AIX(3)



```
-n Is this node channel attached to MVS ('N' if network node)? [Y]


Answer YES
```

Figure 46. PDM Installation Menu on AIX(4)

The messages indicating completion is shown in Figure 47.



```
Saving old environment file in /usr/lpp/dme/config/environment.old
```

Figure 47. IBM exPress Data Mover Installation Steps on AIX

Then the configuration file is edited. You need to modify some parameters. The parameters that probably need to be modified are indicated here with arrows.

```
########################################################
# @(#)env.head  7.1 24 Jun 1997
# Environment Values for PDM
# (You will probably need to edit lines with ′***′)
########################################################
# Basic Parameters that you must set or change
########################################################
### Debug level (0-9) - Set to 5 until working then drop to 0
DebugLevel=5
### The 4 char SMF id of the MVS system.
TargetHost=3090   <---------------- see (1)
```

```
### For PDM network-attached node, uncomment and provide name of the PDM node router
#SessionHost=
### See installation manual - Communication route should be first
two subchannels
Route=3090,0:0  <------------------ see (2)
     or
Route=3090,/dev/rmt4:/dev/rmt5  <-- see (2)
### Remaining subchannels are for data transfer
Device=3090:0902,0:2 <------------- see (2)
Device=3090:0903,0:3 <-------------
Device=3090:0904,0:4 <-------------
Device=3090:0905,0:5 <-------------
Device=3090:0906,0:6 <-------------
Device=3090:0907,0:7 <-------------
      or
Device=3090:0910,/dev/rmt6 <------- see (2)
Device=3090:0911,/dev/rmt7 <-------

Membuffers=4 (to be uncommented for SCSI attachments only)
########################################################
# Basic Parameters that you may optionally change
########################################################
# The default is to /tmp/dme.log
# EdmLog=/var/log/dme.log
# Log file name for Node Router info.
# EdmRouteLog=/var/log/dmeroute.log
########################################################
# Advanced Parameters *** DO NOT CHANGE WITHOUT UNDERSTANDING THEM ***
########################################################
# Define the hostname that we should use for system with multiple
# network interfaces.  Set this to the fastest "network" name if not
# the same as "uname -n"
# HostName=xxx
# FileSize=62
# NetSize=14
# OtherSize=30
# How many MVS tape drive units to allocate
# TapeUnits=1 <------------ see (3)
# Communication and data transfer timing parameters
# SessionTimeout=300
# IdleTime=15
# RetryTime=60
# ResetTime=300
# CommTime=15
# InitIoTime=15
# MaxIoTime=900
# TapeTimeout=900
# User access restriction <----- see (4)
# AllowUser=username
# DisallowUser=username
# FTP Server configuration
# NativeFtpd=/usr/sbin/in.ftpd
# PassiveFtpd=YES
# Task definitions: <task name>,<task path>  - DO NOT CHANGE
Task=DMES,/usr/dme/bin/dmexfer
Task=DMEGW,/usr/dme/bin/dmeserv
```

Notes:

(1) Fill with the SMFID of your default OS/390 server

(2) When asked for the subchannel addresses, give the specification according to the following rule:
<SMFID>,<adapter number>:<subchannel number>

SMFID identifies which OS/390 server is connected to specified subchannel number. The adapter number is a decimal number defined by AIX and can be found in the smit menu ″Show all defined ESCON channel adapters″ (Figure 38 on page 70) if you installed an ESCON adapter, or in ″List all defined Tape Drives″ (Figure 42 on page 71).

These are examples of ESCON specifications:

- 0:0 = Adapter 0 (escon0) subchannel 0
- 0:1 = Adapter 0 (escon0) subchannel 1
- 1:10 = Adapter 1 (escon1) subchannel 10

When you describe *x : y* as a communication path, the next subchannel in sequence (*x : y+1*) is automatically reserved as a second communication path, eliminating the need for you to describe it in the environment file. Since the ESCON subchannels are created using smit, all data paths are created in sequence (y+2, y+3 ...)

Examples of SCSI specifications are:

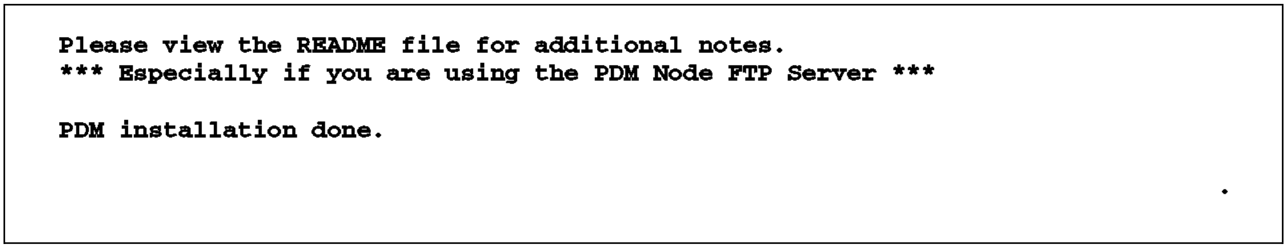
- /dev/rmt4:/dev/rmt5
- /dev/rmt6
- /dev/rmt7

rmt4 and rmt5 tape devices are the communication paths. The other tape devices are the data paths.

(3) The *TapeUnits* parameter needs to be specified, if you intend to send data from the UNIX or Windows NT server directly to tapes on MVS. It represents the maximum number of tape drives that you can allocate simultaneously. Leave the default value for the moment.

(4) Any user can access the IBM exPress Data Mover, unless you specify it differently using the *AllowUser* or *DisallowUser* parameters. Leave the default value as long as you are not in a production process.

The installation process then ends as shown in Figure 48.



```
Please view the README file for additional notes.
*** Especially if you are using the PDM Node FTP Server ***

PDM installation done.
```

Figure 48. IBM exPress Data Mover: End of Installation on AIX

7.1.4.2 AIX Files

The *install_pdm* program has created the following directories :

- /usr/opt/dme
- /usr/opt/dme/bin
- /usr/opt/dme/config

The editable environment files reside in /usr/opt/dme/config.

7.1.4.3 Additional Notes

After you install the IBM exPress Data Mover, add an entry to the /etc/inittab file as follows:

```
pdm:2:once/usr/opt/dme/bin/start_pdm
```

This allows the PDM node router to start automatically at system startup.

In AIX the size of files to be used are limited by the specification in the /etc/security/limits file/. For example, a specification of ″-1″ allows a user to handle any file size. Please refer to AIX documentation to enable large file support.

7.2 HP/UX Installation Planning

This part describes the HP/UX system requirements, provides step-by-step software installation instructions, and lists files that are unique for the HP/UX version of the PDM software using the SCSI/Channel adapter. It continues with the specific information needed to prepare for PDM software installation in the HP/UX environment.

7.2.1 Prerequisites

Before installing the PDM software, make sure that the following system prerequisites are satisfied:

- All Nodes: HP-UX Version 10.2 or 10.20
- SCSI Connection: Before proceeding, refer to SCSI adapters documents regarding installation and configuration of the adapter.

7.2.2 Install the IBM exPress Data Mover

IBM exPress Data Mover is shipped in tar compressed format on either floppy disk or tape. To install the product, perform the following steps :

1. Log in as root.
2. Create the *pdmsys* user group and *pdmsys* user account, so that they own and control the PDM Unix client and data files. No specific privilege is required other than being accessible from any other normal user account.
3. Install the software:

   The PDM software is shipped in a package, which is installed using the software install tool *swinstall*. You may want to use the *swcopy* utility to first copy the software package to an installation depot, depending on your system administration policies.

Attention: When installing a PDM software upgrade to an existing PDM system, back up any parameter files before downloading the new upgrade to prevent existing parameter files from being overwritten.

- *cd /opt/pdm*
- *swinstall PDM*

7.2.3 SCSI Adapter I/O Definition

Before proceeding, refer to SCSI adapters documents regarding installation and configuration of the adapter.

7.2.3.1 SCSI Connection

The SCSI adapter is defined on the SCSI bus as a tape device with a specific target address, and it consists of several logical unit numbers (LUNs) for that target (normally 8). Consult your system administration manuals for additional information regarding the connection and configuration of tape drives.

After the SCSI adapter is recognized by HP/UX, verify that the proper device names have been created to support multiple LUNs for the devices. Displayed is a variety of file names in the /dev/rmt directory. In particular, the devices named as *c#t#d#BEST*, are displayed, where:

- c# is the controller number
- t# is the SCSI target address for the SCSI adapter
- d# is the LUN

If the only device name created is for LUN 0 (that is, *c0t4d0BEST*), create the remaining devices using the *mknod* command and set their access permissions to be read/write for everyone as follows:

- *mknod c#t#d#BEST c 205 0x####00*
- *chmod 666 c#t#d#BEST*

You can identify the proper device name based on the device name for the logical tape device name on LUN 0 that was created by the system. The major device number is always 205 for HP/UX. Verify that the major device number is correct. For the minor device number, 0x####00, match the existing minor device number for LUN 0. Change the fourth of the six hexadecimal digits to be the LUN number.

For example, for a SCSI adapter connected to controller 0 at SCSI target address 4, execute the following commands to define all eight LUNs:

```
# cd /dev/rmt
# ls -l *BEST
crw-rw-rw-  1  root  root   205  0x004000  Jan  01  12:00  c0t4d0BEST
#  mknod  c0t4d1BEST  c  205  0x004100
#  mknod  c0t4d2BEST  c  205  0x004200
 ....
#  mknod  c0t4d7BEST  c  205  0x004700
#  chmod  666  c0t4d*BEST
```

During the configuration of PDM (when specifying the configuration in the PDM environment file) use the */dev/rmt/c#t#d#BEST* device name.

7.2.3.2 Hardware Verification

After the hardware is connected and brought online to OS/390, test basic connectivity to each of the SCSI tape device names by issuing a tape rewind for each device using the *mt* command:

- *mt -t /dev/rmt/c0t4d0BEST rewind*

Do not attempt to start the PDM software until this command completes with no errors.

7.2.4 Configure IBM exPress Data Mover

The first steps are to install the software and configure the adapters.

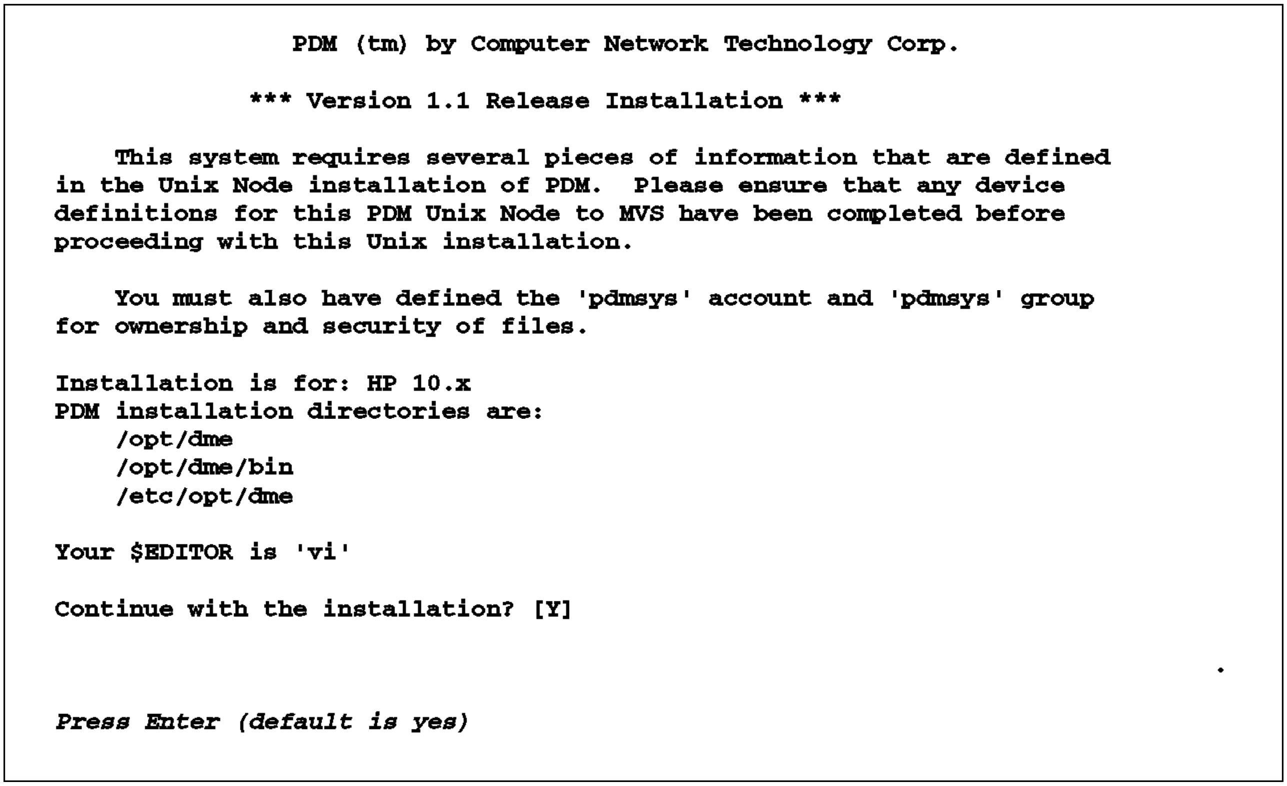
7.2.4.1 Run the Installation Program

Once the software is installed and the adapters have been correctly configured, log in as root and run the installation program:

- *cd /opt/dme*
- *./install_pdm*

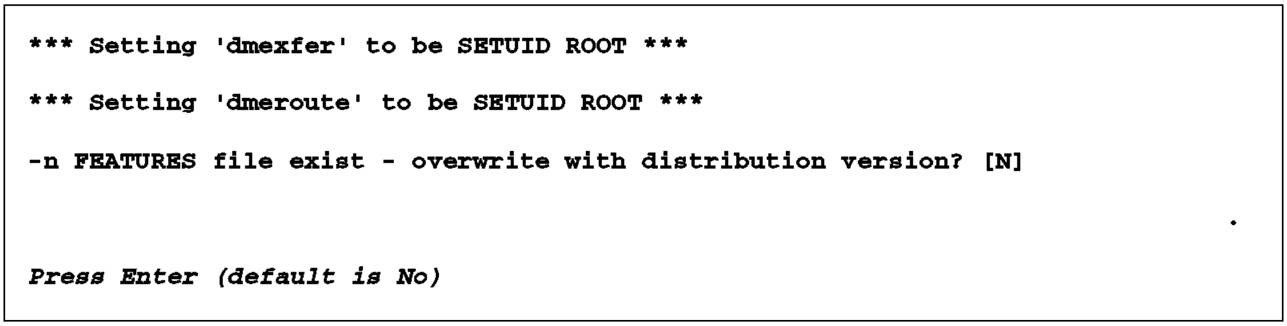
This process creates all of the necessary files and directories required by the PDM components, and it assists in the configuration of the environment file.

Figure 49, Figure 50 on page 79, Figure 51 on page 79, and Figure 52 on page 79 show various prompts.



```
                 PDM (tm) by Computer Network Technology Corp.

              *** Version 1.1 Release Installation ***

        This system requires several pieces of information that are defined
    in the Unix Node installation of PDM.  Please ensure that any device
    definitions for this PDM Unix Node to MVS have been completed before
    proceeding with this Unix installation.

        You must also have defined the 'pdmsys' account and 'pdmsys' group
    for ownership and security of files.

    Installation is for: HP 10.x
    PDM installation directories are:
        /opt/dme
        /opt/dme/bin
        /etc/opt/dme

    Your $EDITOR is 'vi'

    Continue with the installation? [Y]


    Press Enter (default is yes)
```

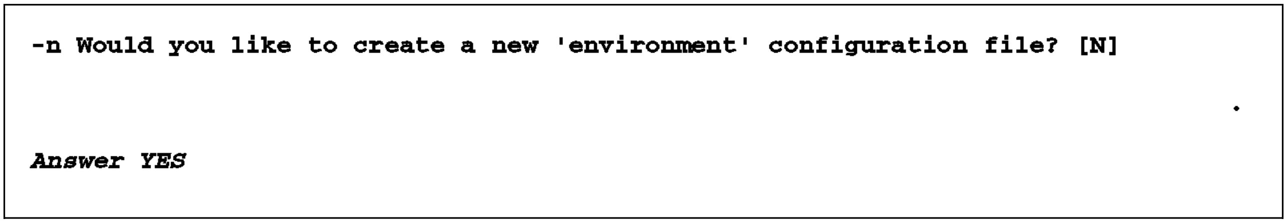
Figure 49. PDM Installation Menus on HP(1)



```
*** Setting 'dmexfer' to be SETUID ROOT ***

*** Setting 'dmeroute' to be SETUID ROOT ***

-n FEATURES file exist - overwrite with distribution version? [N]
```

Press Enter (default is No)

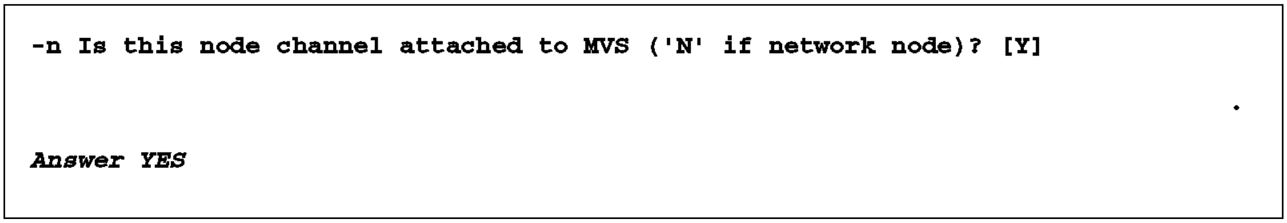
Figure 50. PDM Installation Menus on HP(2)



```
-n Would you like to create a new 'environment' configuration file? [N]
```

Answer YES

Figure 51. PDM Installation Menus on HP(3)

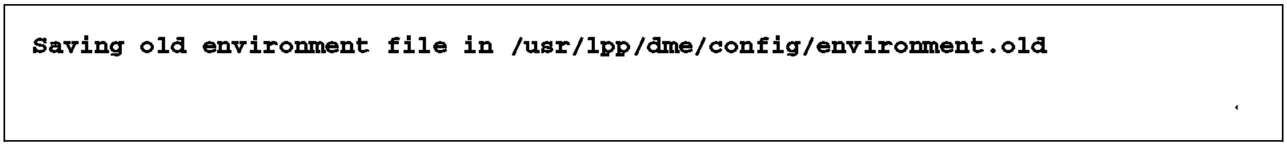


```
-n Is this node channel attached to MVS ('N' if network node)? [Y]
```

Answer YES

Figure 52. PDM Installation Menus on HP(4)

The completion of the configuration results in the message shown in Figure 53.



```
Saving old environment file in /usr/lpp/dme/config/environment.old
```

Figure 53. PDM Configuration File Completed for HP/UX

Then the configuration file is edited. We show here only the parameters that may need to be modified. For a complete view of the configuration file and for explanations on the parameters which you can modify, please refer to 7.1.4, "Configure IBM exPress Data Mover" on page 72.

```
########################################################
# @(#)env.head  7.1 24 Jun 1997
# Environment Values for PDM
# (You will probably need to edit lines with '***')
########################################################
# Basic Parameters that you must set or change
########################################################
### Debug level (0-9) - Set to 5 until working then drop to 0
DebugLevel=5
### The 4 char SMF id of the MVS system.
TargetHost=3090   <---------------- (1)
### For PDM network-attached node, uncomment and provide name of the PDM node router
```

```
#SessionHost=
### See installation manual - Communication route should be first
two subchannels
Route=3090,/dev/rmt/c0t4d0BEST:/dev/rmt/c0t4d1BEST <--(2)
### Remaining subchannels are for data transfer
Device=3090:0916,/dev/rmt/c0t4d2BEST <-------(2)
Device=3090:0917,/dev/rmt/c0t4d3BEST <-------(2)

Membuffers=4

########################################################
# Basic Parameters that you may optionally change
########################################################
 .........
########################################################
# Advanced Parameters *** DO NOT CHANGE WITHOUT UNDERSTANDING THEM ***
########################################################
# ......
# How many MVS tape drive units to allocate
# TapeUnits=1 <------------(3)
# Communication and data transfer timing parameters
# ........
# User access restriction <-----(4)
# AllowUser=username
# DisallowUser=username
# FTP Server configuration
# NativeFtpd=/usr/sbin/in.ftpd
# PassiveFtpd=YES
# Task definitions: <task name>,<task path>  - DO NOT CHANGE
 .......
```

Notes:

(1) Fill with the SMFID of your default OS/390 server.

(2) When asked for the device, give the SCSI device name that correspond to both the communication paths and the data transfer paths. Communications paths are described in the Route line, Data paths are described in the Device lines. Paths are created in sequence (y+2, y+3 ...). The SMFID identifies which OS/390 server is connected to specified path.

Examples of SCSI specifications are :

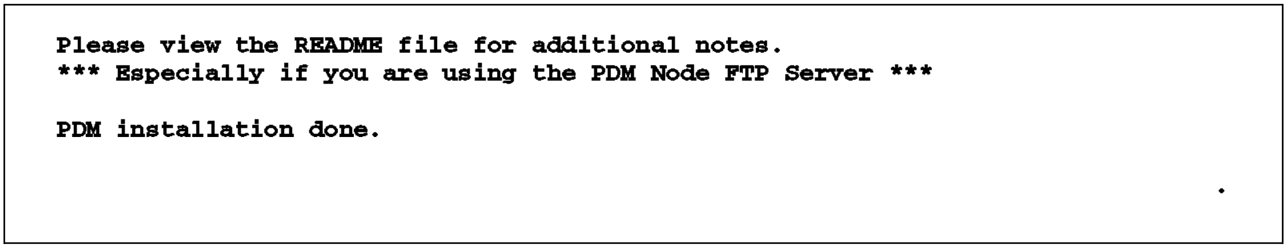
- /dev/rmt/c0t4d0BEST
- /dev/rmt/c0t4d1BEST
- /dev/rmt/c0t4d2BEST
- /dev/rmt/c0t4d3BEST

/dev/rmt/c0t4d0BEST and /dev/rmt/c0t4d1BEST tape devices are communication paths. The other tape devices are the data paths.

(3) The *TapeUnits* parameter needs to be specified, if you intend to send data from the UNIX or Windows NT server directly to tapes on MVS. It represents the maximum number of tape drives that you can allocate simultaneously. Leave the default value for the moment.

(4) Any user can access the IBM exPress Data Mover, unless you specify it differently using the *AllowUser* and/or the *DisallowUser* parameters. Leave the default value as long as you are not in a production process.

The installation process then ends, as shown in Figure 54 on page 81.



```
Please view the README file for additional notes.
*** Especially if you are using the PDM Node FTP Server ***

PDM installation done.
```

Figure 54. IBM exPress Data Mover: End of Installation on HP/UX

7.2.4.2 HP/UX Files

After the software is extracted and the installation script has completed, the following files and directories exist:

File or Directory	Description
/opt/dme	Base directory for all installed files
/opt/dme/install_pdm	Installation script.
/opt/dme/README	File containing installation information and known problems
/etc/opt/dme/environment	File containing the definition of various parameters to the PDM system
/etc/opt/dme/features	File containing flags specifying which features have been purchased.

7.2.4.3 Additional Notes

We strongly recommend that you start the node router automatically at system startup. After installing the PDM software, edit the */etc/rc.local* file to automatically execute */usr/dme/bin/start_pdm* at system startup.

7.3 Solaris Installation Planning

This section describes the Solaris system requirements, provides step-by-step software installation instructions, and lists files that are unique for that version of the PDM software using the SCSI adapter. It continues with specific information needed to prepare for PDM software installation in the Solaris environment.

7.3.1 Prerequisites

Before installing the PDM software, make sure that the following system prerequisites are satisfied:

- All nodes are running Solaris Version 2.4, 2.5, or 2.5.1.
- SCSI Adapter is installed.

7.3.2 Software Installation

The PDM software is shipped in *pkgadd* format.

Attention: When installing a PDM software upgrade to an existing PDM system, back up any parameter files before downloading the new upgrade to prevent existing parameter files from being overwritten. The steps are these:

1. Log in as root.

2. Create the *pdmsys* user group and *pdmsys* user account, so that they own and control the PDM UNIX client and data files. No specific privilege is required other than being accessible from any other normal user account.
3. Run pkgadd to install the PDM files:
   - */usr/sbin/pkgadd -d /dev/rmt/x*

     Select the default for all products.

7.3.3 SCSI Adapter I/O Definition

Before proceeding, refer to SCSI adapters documents regarding installation and configuration of the adapter.

7.3.3.1 SCSI Definition

The SCSI adapter is defined on the SCSI bus as a tape device with a specific target address. It consists of several logical unit numbers (LUNs) for that target. In order for the Solaris operating system to detect the presence of the additional LUNs, edit the SCSI tape configuration file */kernel/drv/st.conf* and add a specification for the nonzero LUN devices. The unmodified file contains the following:

```
name="st"  class="scsi"      target=0  lun=0;
name="st"  class="scsi"      target=1  lun=0;
name="st"  class="scsi"      target=2  lun=0;
 ...
```

For the target address that you plan to use for the SCSI adapter, duplicate the entry for the additional LUNs. For example, if the SCSI adapter is at SCSI address 4, after the target=4 lun=0 entry, add seven more for LUN 1 through LUN 7 as follows :

```
name="st"  class="scsi"      target=4  lun=1;
name="st"  class="scsi"      target=4  lun=2;
 ...
name="st"  class="scsi"      target=4  lun=7;
```

Additional entries must not conflict with any of the existing tape drives that are already connected to the system. After the st.conf file has changed, rebuild the device definitions for the operating system. Consult your system administrators manual for directions. The simplest way to do this is to create a dummy file named */reconfigure* and then restart the system:

```
#  cd  /
#  touch  /reconfigure
#  sync
#  reboot
```

After the system restarts, examine the /dev/rmt device directory for the tape devices that are simple tape numbers (0, 1, 2, 3, 4). The directory entry will be a symbolic link to the actual system device. For example,

```
#  cd  /dev/rmt
#  ls  -l  0
lrwxrwxrwx  1  root  root  50  Apr  16  11:16  0  ->
 ../../devices/sbus@1,f8000000/esp@0,800000/st@4,0:
#  ls  -l  1
lrwxrwxrwx  1  root  root  50  Apr  16  11:16  1  ->
 ../../devices/sbus@1,f8000000/esp@0,800000/st@4,1:
```

Notice that the last part of the name is the SCSI target number and LUN. Therefore *st@4,1* is SCSI target 4, LUN 1.

It is very important that the exact SCSI device names are mapped to the correct subchannel address. Do not assume that device */dev/rmt/0* is LUN 0. However, all LUNs for a given SCSI target address should be sequential.

During the configuration of PDM, configure subchannel addresses to PDM. When doing this, provide a device specification. This specification is the tape device name. For example */dev/rmt/1*.

7.3.3.2 Hardware Verification

After the hardware is connected and brought online to OS/390, test basic connectivity to each of the SCSI tape device names by issuing a tape rewind to each device using the mt command:

- *mt -f /dev/rmt/0 rewind*

Do not attempt to start the PDM software until this command completes with no errors.

7.3.4 Configure IBM exPress Data Mover

The first steps are to install the software and configure the adapters.

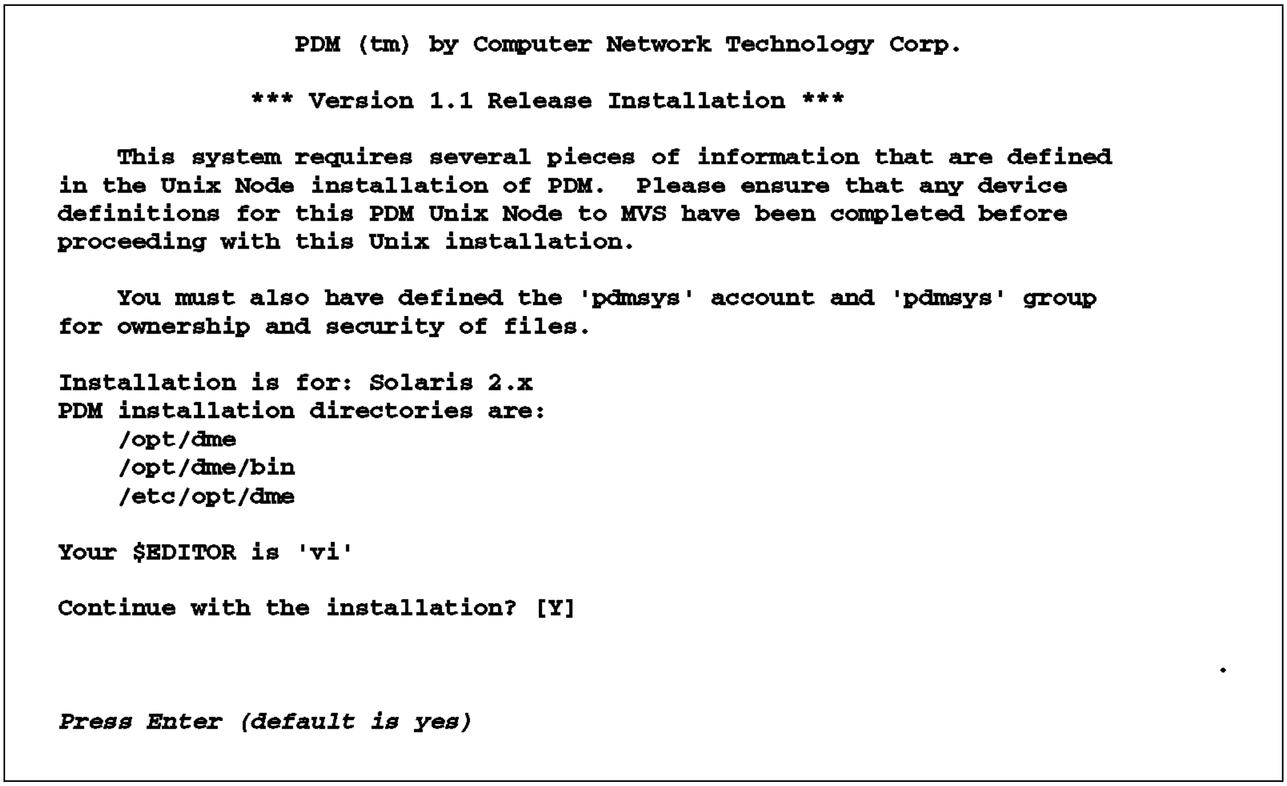
7.3.4.1 Run the Installation Program

Once the software is installed and the adapters have been correctly configured, log in as root and run the installation program :

- *cd /opt/dme*
- *./install_pdm*

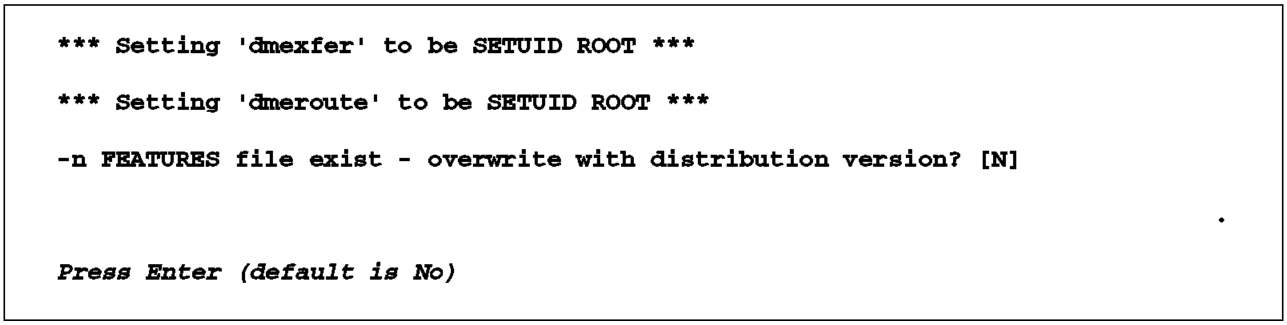
This process creates all of the necessary files and directories required by all the PDM components, and it assists in the configuration of the environment file.

Figure 55 on page 84 shows the first promt. Figure 56 on page 84, Figure 57 on page 84, and Figure 58 on page 85 show the subsequent prompts.



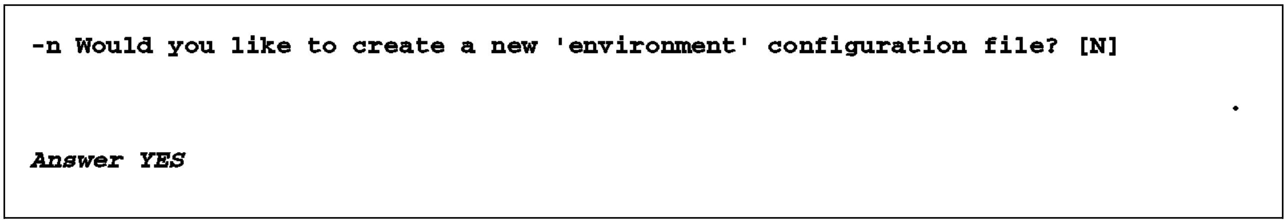
```
                   PDM (tm) by Computer Network Technology Corp.

                *** Version 1.1 Release Installation ***

        This system requires several pieces of information that are defined
    in the Unix Node installation of PDM.  Please ensure that any device
    definitions for this PDM Unix Node to MVS have been completed before
    proceeding with this Unix installation.

        You must also have defined the 'pdmsys' account and 'pdmsys' group
    for ownership and security of files.

    Installation is for: Solaris 2.x
    PDM installation directories are:
        /opt/dme
        /opt/dme/bin
        /etc/opt/dme

    Your $EDITOR is 'vi'

    Continue with the installation? [Y]


    Press Enter (default is yes)
```

Figure 55. PDM Installation Menu on Sun(1)



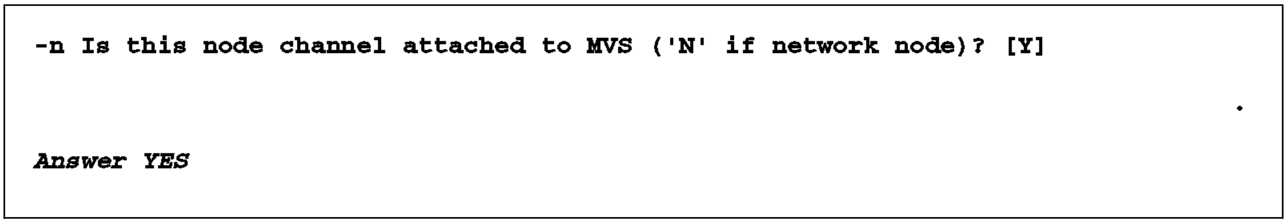
```
    *** Setting 'dmexfer' to be SETUID ROOT ***

    *** Setting 'dmeroute' to be SETUID ROOT ***

    -n FEATURES file exist - overwrite with distribution version? [N]


    Press Enter (default is No)
```

Figure 56. PDM Installation Menu on Sun(2)



```
  -n Would you like to create a new 'environment' configuration file? [N]


  Answer YES
```

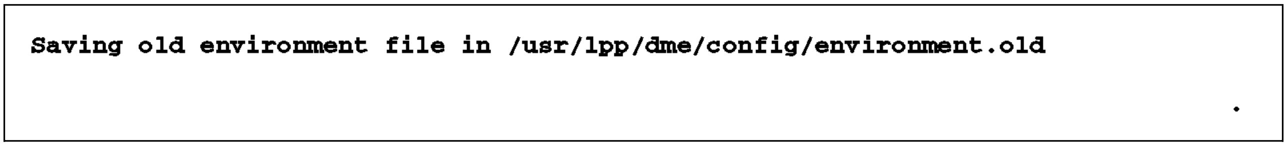
Figure 57. PDM Installation Menu on Sun(3)



```
-n Is this node channel attached to MVS ('N' if network node)? [Y]

Answer YES
```

Figure 58. PDM Installation Menu on Sun(4)

The messages shown in Figure 59 indicates that the configuration file is completed.



```
Saving old environment file in /usr/lpp/dme/config/environment.old
```

Figure 59. PDM Configuration File Completed for Sun Solaris

Then the configuration file is edited. Here we show only the parameters that may need to be modified. For a complete view of the configuration file and for explanations of the parameters you can modify, please refer to 7.1.4, “Configure IBM exPress Data Mover” on page 72.

```
########################################################
# @(#)env.head  7.1 24 Jun 1997
# Environment Values for PDM
# (You will probably need to edit lines with ′***′)
########################################################
# Basic Parameters that you must set or change
########################################################
### Debug level (0-9) - Set to 5 until working then drop to 0
DebugLevel=5
### The 4 char SMF id of the MVS system.
TargetHost=3090         <-------------(1)
### For PDM network-attached node, uncomment and provide name of the PDM node router
#SessionHost=
### See installation manual - Communication route should be first
two subchannels
Route=3090,/dev/rmt/17:/dev/rmt/31 <--(2)
### Remaining subchannels are for data transfer
Device=3090:091A,/dev/rmt/32 <-------(2)
Device=3090:091B,/dev/rmt/33 <-------(2)

Membuffers=4

########################################################
# Basic Parameters that you may optionally change
########################################################
 .........
########################################################
# Advanced Parameters *** DO NOT CHANGE WITHOUT UNDERSTANDING THEM ***
########################################################
# ......
# How many MVS tape drive units to allocate
# TapeUnits=1 <------------(3)
# Communication and data transfer timing parameters
# .......
# User access restriction <-----(4)
# AllowUser=username
# DisallowUser=username
# FTP Server configuration
# NativeFtpd=/usr/sbin/in.ftpd
# PassiveFtpd=YES
# Task definitions: <task name>,<task path>  - DO NOT CHANGE
 .......
```

Notes:

(1) Fill with the SMFID of your default OS/390 server.

(2) When asked for the device, give the SCSI device name that corresponds to both the communication paths and the data transfer paths. Communications paths are described in the Route line, Data paths are described in the Device lines. Paths are created in sequence (y+2, y+3 ...). The SMFID identifies which OS/390 server is connected to specified path.

Example of SCSI specifications are:

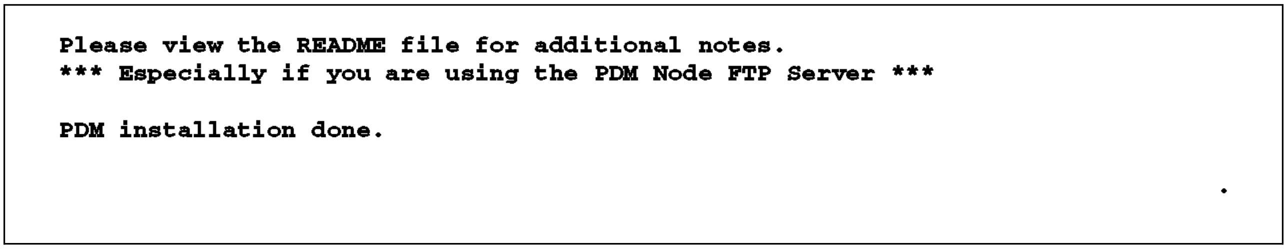
- /dev/rmt/17
- /dev/rmt/31
- /dev/rmt/32
- /dev/rmt/33

/dev/rmt/17 and /dev/rmt/31 tape devices are the communication paths. The other tape devices are the data paths.

(3) The *TapeUnits* parameter needs to be specified, if you intend to send data from the UNIX or Windows NT server directly to tapes on MVS. It represents the maximum number of tape drives that you can allocate simultaneously. Leave the default value for the moment.

(4) Any user can access the IBM exPress Data Mover, unless you specify it differently using the *AllowUser* or *DisallowUser* parameter. Leave the default value as long as you are not in a production process.

The installation process then ends, as shown in Figure 60.



```
Please view the README file for additional notes.
*** Especially if you are using the PDM Node FTP Server ***

PDM installation done.
```

Figure 60. IBM exPress Data Mover: End of Installation on Sun Solaris

7.3.4.2 Solaris File Descriptions

After the software is extracted and the installation script has completed, the following Solaris-specific files and directories exist:

File or Directory	**Description**
/opt/dme	Base directory for all installed PDM files
/opt/dme/install_pdm	Installation script.
/opt/dme/README	File containing installation information and known problems

7.3.4.3 Additional Notes

We strongly recommend that the system be configured in a way that the node router starts automatically at system startup. After installing the PDM software, edit the sample file */opt/dme/config/dme_init.d*, and perform any customization necessary regarding the initialization of pdmroute. Copy it into *etc/init.d/dme* and make the correct link into the the proper multiuser run level directory; normally, */etc/rc2.d/S99pdm*.

7.4 Windows NT Installation Planning

This section describes the Microsoft Windows NT system requirements, provides software installation instructions, and lists files that are unique for the NT version of the PDM software using the SCSI adapter in the NT environment.

7.4.1 Prerequisites

Before installing the PDM software, make sure the following system prerequisites are satisfied.

- All Nodes: Microsoft Windows NT, Version 4.0
- Physical Attachment: SCSI Adapter

Since the PDM configuration starts automatically after the software installation is completed, we suggest that you first go through the SCSI definition, and then install the software.

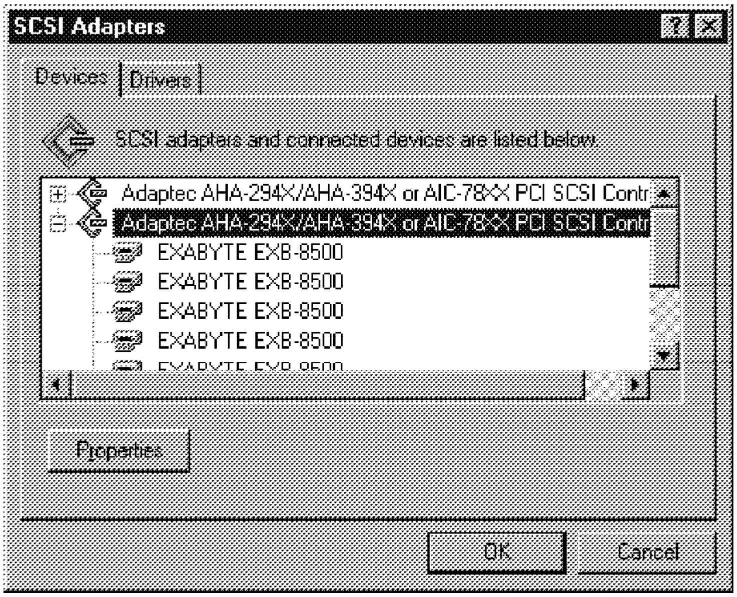
7.4.2 Windows NT SCSI Adapter I/O Definition

Before proceeding, refer to SCSI adapters documents regarding installation and configuration of the adapter.

The SCSI adapter is defined on the SCSI bus as an Exabyte EXB-8500 tape device with a specific target address, and it consists of several logical unit numbers (LUNS) for that target (normally eight). If not already installed, you have to install the Exabyte2 device driver that comes with your Windows NT CD-ROM.

Power off the Windows NT system and attach the SCSI cable to the appropriate adapter card. Power on the system and bring Windows NT to an operational state. During the power-on cycle, Windows NT should have recognized the IBM InfoSpeed Data Gateway and defined the devices.

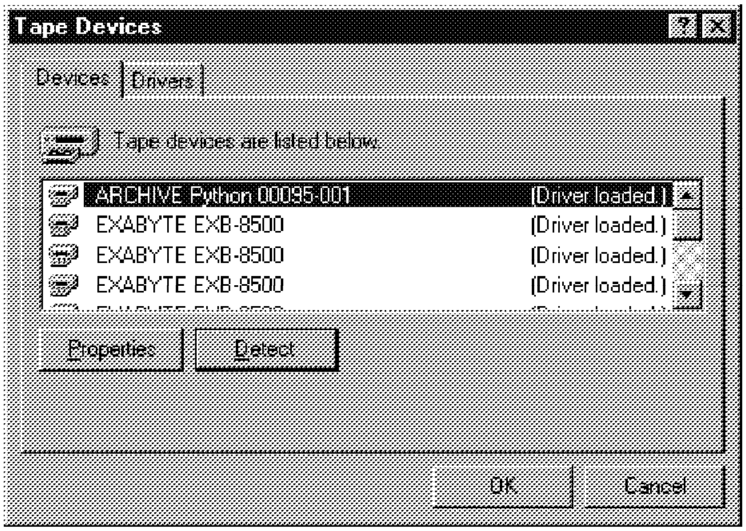
Using the Tape Devices control panel, display the attached tape devices and assigned tape drivers to verify the definition, as shown in Figure 61 on page 88 through Figure 64 on page 89. Windows NT should have defined each of the eight tape devices as EXB-8500 tape devices using the Exabyte2 tape driver, which comes standard with NT. Click on *Properties* for each of the tape devices to verify proper SCSI target ID and LUN.



SCSI adapters and connected devices are listed below.

```
⊞ Adaptec AHA-294X/AHA-394X or AIC-78XX PCI SCSI Contr
⊟ Adaptec AHA-294X/AHA-394X or AIC-78XX PCI SCSI Contr
     EXABYTE EXB-8500
     EXABYTE EXB-8500
     EXABYTE EXB-8500
     EXABYTE EXB-8500
     EXABYTE EXB-8500
```

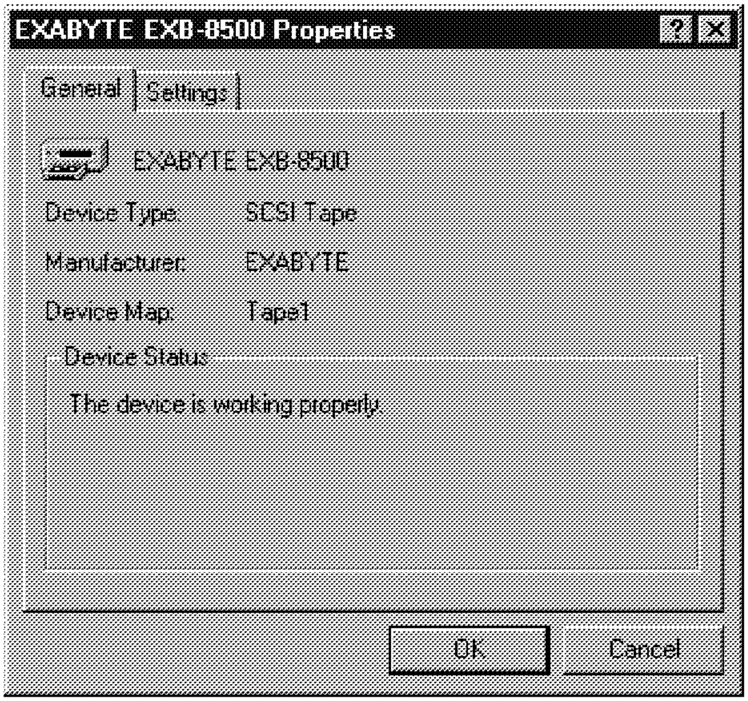
Figure 61. SCSI Channel Definition on Windows NT (1)



Tape devices are listed below.

```
ARCHIVE Python 00095-001        (Driver loaded.)
EXABYTE EXB-8500                (Driver loaded.)
EXABYTE EXB-8500                (Driver loaded.)
EXABYTE EXB-8500                (Driver loaded.)
```

Figure 62. SCSI Channel Definition on Windows NT (2)



EXABYTE EXB-8500 Properties

EXABYTE EXB-8500

Device Type: SCSI Tape

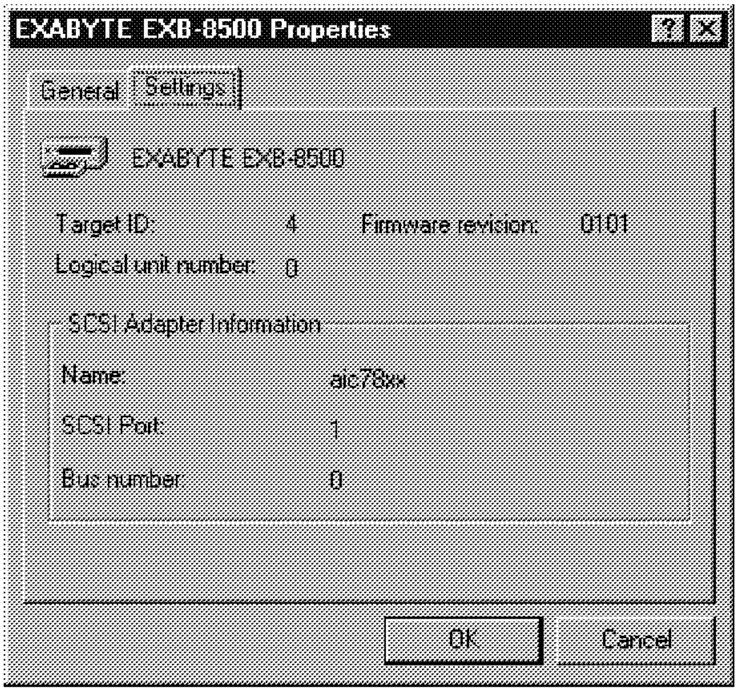
Manufacturer: EXABYTE

Device Map: Tape1

Device Status

The device is working properly.

Figure 63. SCSI Channel Definition on Windows NT (3)



EXABYTE EXB-8500 Properties

EXABYTE EXB-8500

Target ID: 4 Firmware revision: 0101

Logical unit number: 0

SCSI Adapter Information

Name: aic78xx

SCSI Port: 1

Bus number: 0

Figure 64. SCSI Channel Definition on Windows NT (4)

The PDM configuration utility provides an easy to use interface to select the appropriate tape device numbers for use with PDM. However, the logical tape devices must be properly recognized by the Windows NT tape driver before the PDM configuration utility can detect the existence of the tape devices.

7.4.3 Install the IBM exPress Data Mover

The PDM software is shipped on a floppy disk. Install the PDM software by executing *setup.exe* as seen in Figure 67 on page 91. After completing the installation of the software, it automatically executes the pdmcfg.exe program, which configures the PDM software for use with SCSI adapters.

Attention: When installing a PDM software upgrade onto an existing PDM system, back up any parameter files before downloading the new upgrade to prevent existing parameter files from being overwritten.

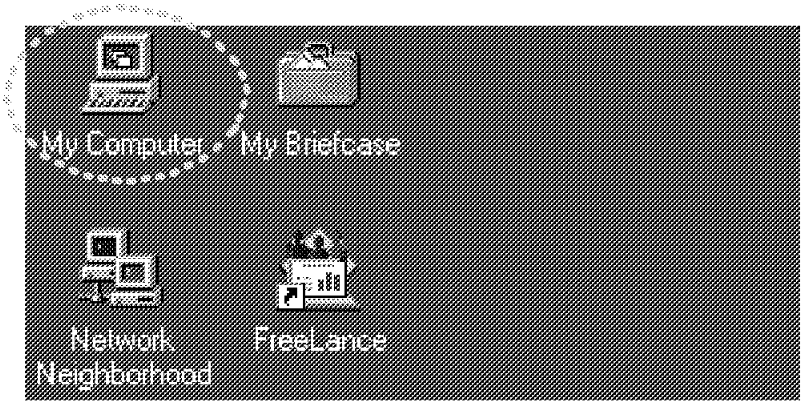


Figure 65. PDM Installation Menu on Windows NT (1)

Insert the floppy disk and select the icon *My Computer*, which you find on the desktop of your Windows NT system (as shown in Figure 65) and a window for device selection is shown, as in Figure 66.

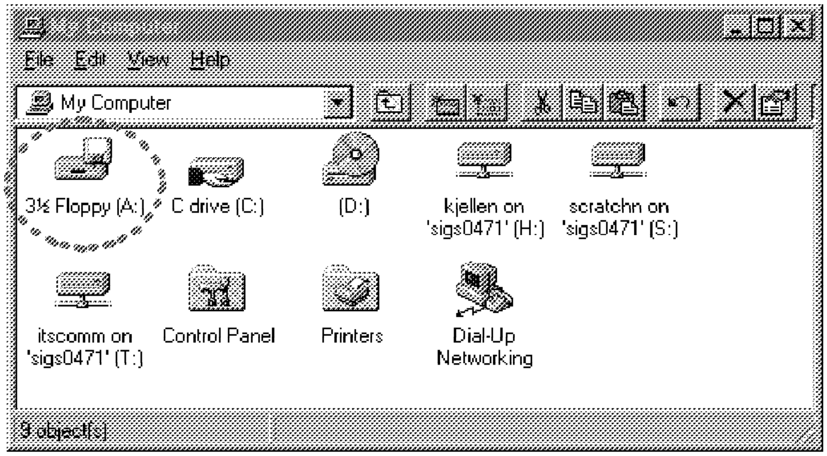


Figure 66. PDM Installation Menu on Windows NT (2)

In the device selection window, you select the device where you have mounted the floppy disk, as indicated in Figure 66. The objects on the floppy disk are then shown as in Figure 67 on page 91.

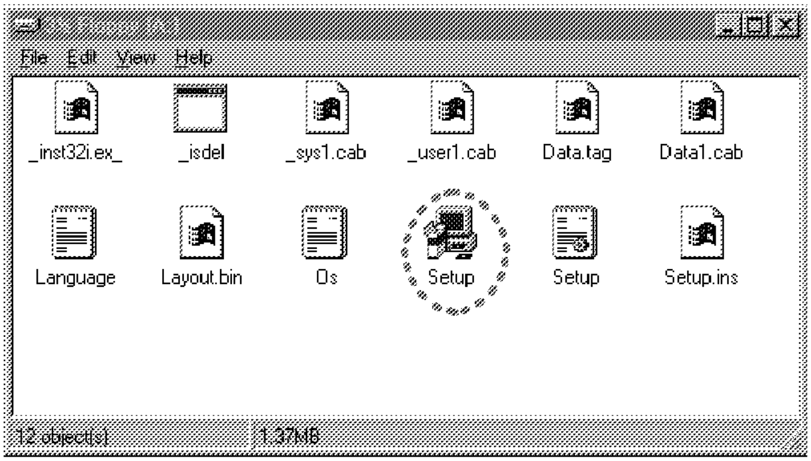
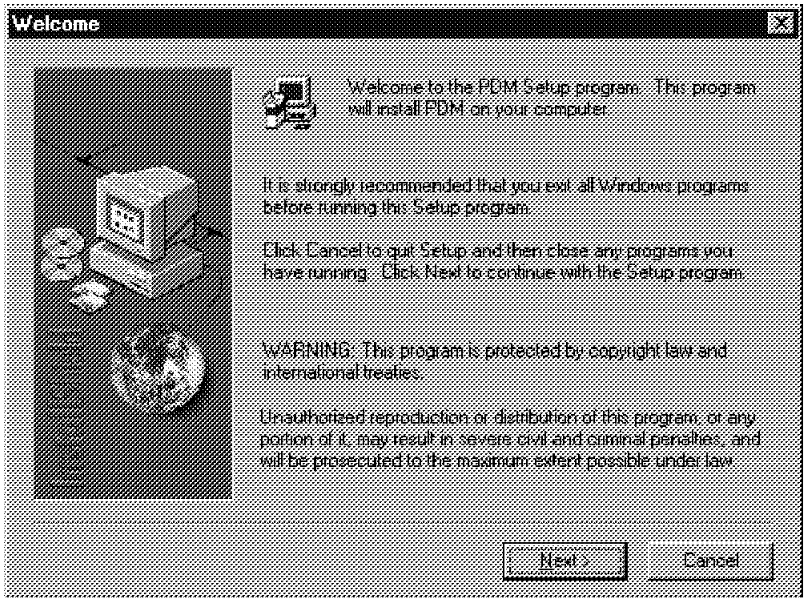


Figure 67. PDM Installation Menu on Windows NT (3)

When you double-click on the icon for the *Setup* (Figure 67), the PDM installation starts and a series of installation windows (Figure 68 through Figure 72 on page 93) are shown.


Welcome

Welcome to the PDM Setup program. This program will install PDM on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

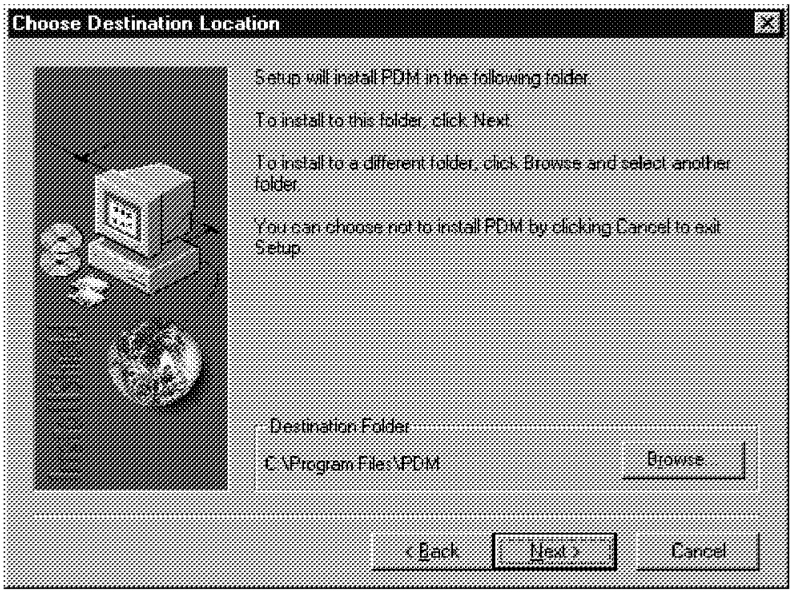
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Figure 68. PDM Installation Menu on Windows NT (4)

The window shown in Figure 68 is a standard setup window shown during installations of programs in Windows NT. Please follow recommendations shown.



Setup will install PDM in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

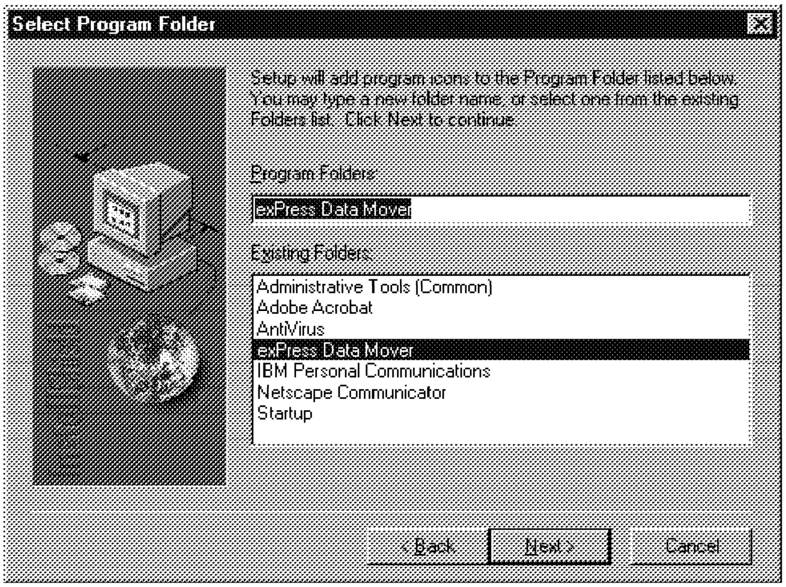
You can choose not to install PDM by clicking Cancel to exit Setup.

Destination Folder

C:\Program Files\PDM

Figure 69. PDM Installation Menu on Windows NT (5)

In the window shown in Figure 69, you can select another destination folder if needed.



Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

Program Folders:

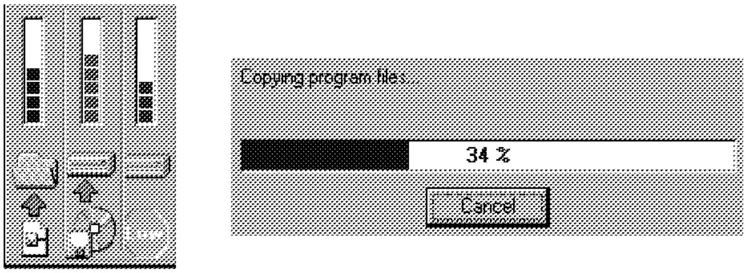
exPress Data Mover

Existing Folders:

Administrative Tools (Common)
Adobe Acrobat
AntiVirus
exPress Data Mover
IBM Personal Communications
Netscape Communicator
Startup

Figure 70. PDM Installation Menu on Windows NT (6)

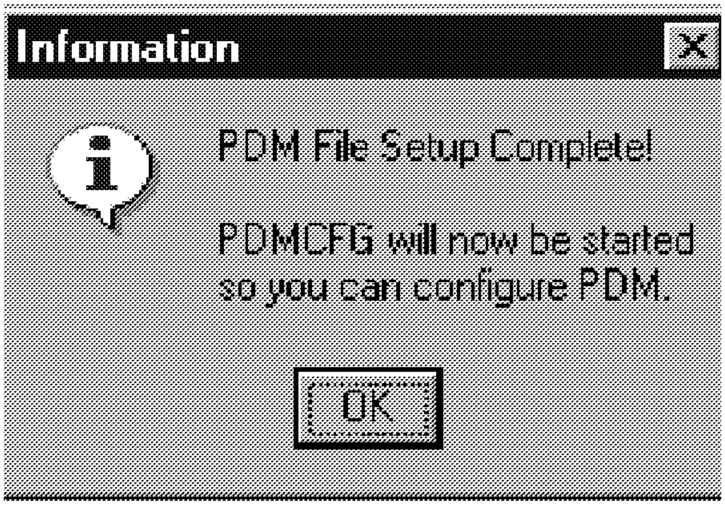
In the window shown in Figure 70, you can select a new folder name rather than accepting the suggested one.


Copying program files...

34 %

Figure 71. PDM Installation Menu on Windows NT (7)

When you select *Next* in the window shown in Figure 70 on page 92, bars indicating percent copied (shown in Figure 71) grow. When all files have been copied onto the hard disk and the *Setup* process has finished you will see an infomation window as in Figure 72.



PDM File Setup Complete!

PDMCFG will now be started so you can configure PDM.

Figure 72. Configuration File Completed for Windows NT (8)

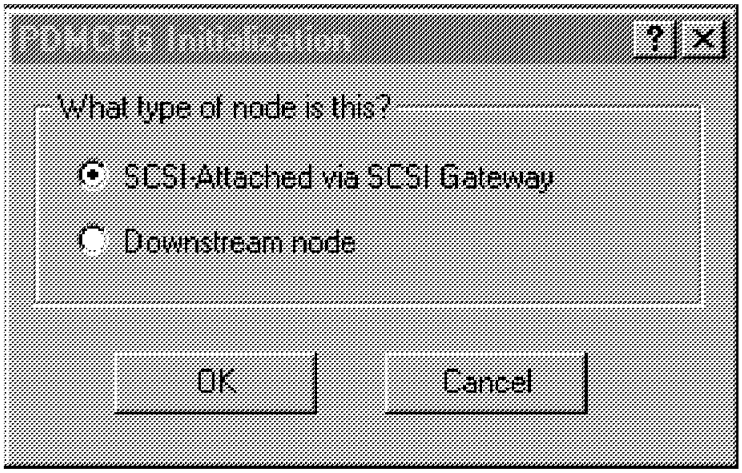
When you click on **OK** the configuration process starts automatically. This process is described in 7.4.4, “Configure IBM exPress Data Mover for Windows NT.”

7.4.4 Configure IBM exPress Data Mover for Windows NT

Once the software is installed, Windows NT runs the installation program automatically.

7.4.4.1 The Configuration Program

The pdmcfg.exe file is the main PDM installation program. It determines the system type, creates directories, sets the required authorities for the PDM services, edits the environment file during the installation process, and asks you to ″fill in the fields.″



PDM CFG Initialization

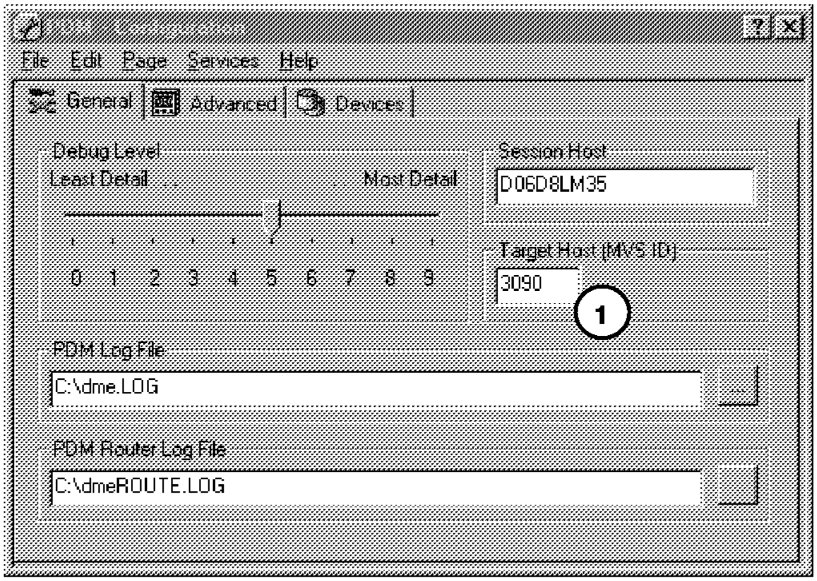
What type of node is this?

● SCSI-Attached via SCSI Gateway

○ Downstream node

Figure 73. PDM Configuration Menu on Windows NT (1)

On the first panel, Figure 73, select the SCSI-attached Gateway. The next few windows, Figure 74 through Figure 77 on page 96, are used to define the environment for IBM exPress Data Mover running under Windows NT. The information to be specified is similar to the information specified in the environmental file used in a UNIX system; see the sample configuration file in 7.1.4, "Configure IBM exPress Data Mover" on page 72.



Debug Level: 5

Session Host: D06D8LM35

Target Host (MVS ID): 3090

PDM Log File: C:\dme.LOG

PDM Router Log File: C:\dmeROUTE.LOG

Figure 74. PDM Configuration Menu on Windows NT (2)

Notes:

(1) On the second panel, Figure 74, in folder *General*, the SMFID of your OS/390 should be specified. The recommendation for the Debug Level, shown in Figure 74, is to set it to 5 until the system is working and then drop to 0.

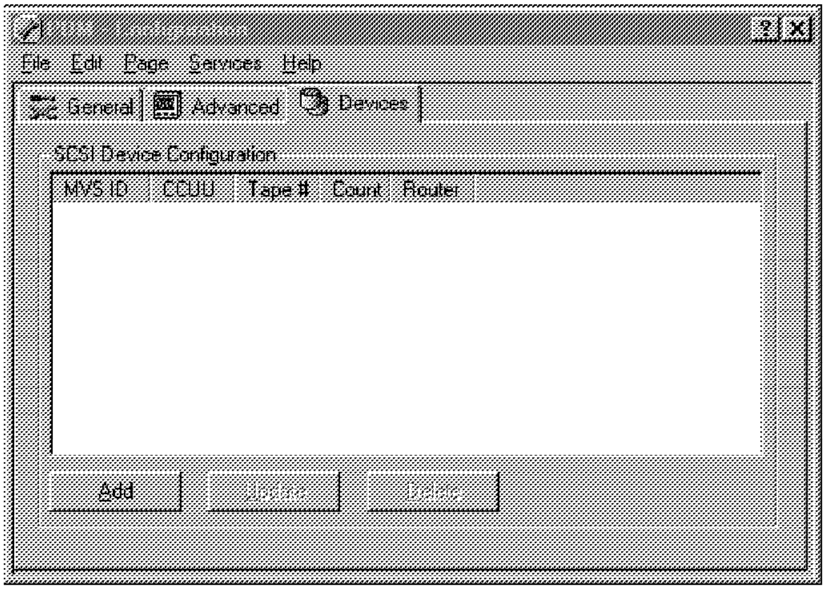
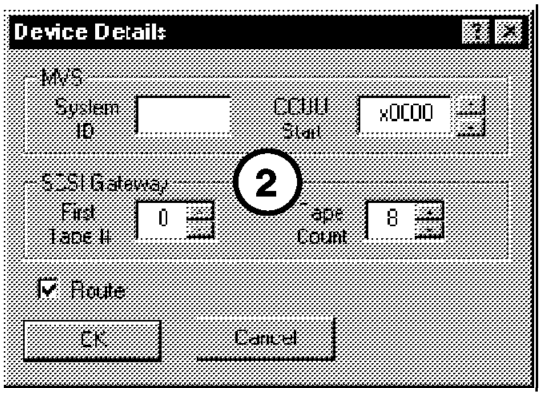


Figure 75. PDM Configuration Menu on Windows NT (3)

On the second panel, Figure 75, in folder *Devices* you specify what corresponds to the Route and Device statements in the UNIX environmental file. When you select **Add** a window like that in Figure 76 is shown. Select **Add** once for every OS/390 (MVS) image attached.



System ID:

CCUU Start: x0C00

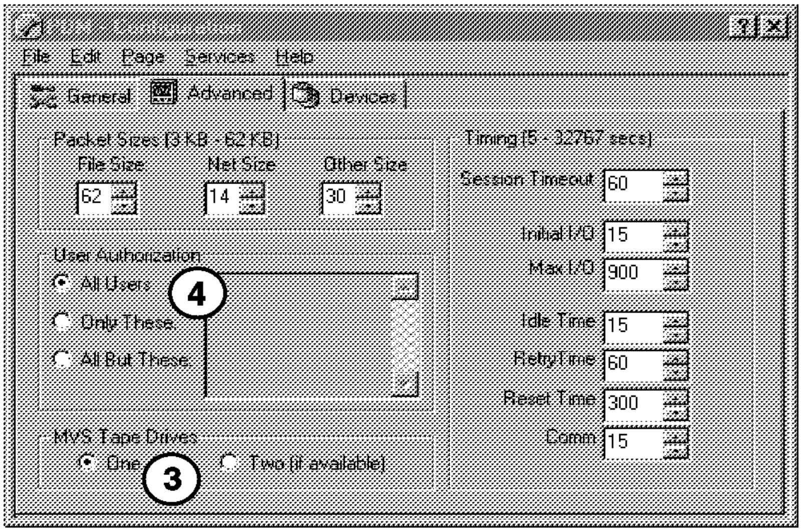
First Tape #: 0

Tape Count: 8

☑ Route

Figure 76. PDM Configuration Menu on Windows NT (4)

(2) In the Device Detail window, Figure 76, specify the SMFID for the OS/390 system, for example 3090. Also specify the device address starting number defined in HCD in the S/390 system, for example x0918, the first tape (must match the appropriate Device Map) corresponding to LUN0 for the Gateway, and the number of devices (number of LUNs, in most cases eight).



Packet Sizes (3 KB - 62 KB)
File Size: 62
Net Size: 14
Other Size: 30

User Authorization
● All Users
○ Only These
○ All But These

MVS Tape Drives
● One
○ Two (if available)

Timing (5 - 32767 secs)
Session Timeout: 60
Initial I/O: 15
Max I/O: 900
Idle Time: 15
RetryTime: 60
Reset Time: 300
Comm: 15

Figure 77. PDM Configuration Menu on Windows NT (5)

(3) On the second panel, Figure 77, in folder *Advanced* you should select the number of tape drives that you can allocate simultaneously in MVS. The MVS Tape Drives option needs to be specified, if you intend to send data from the Windows NT server directly to tapes on MVS.

(4) Specify which users can access the IBM exPress Data Mover. Any user can access the IBM exPress Data Mover, unless you specify otherwise by selecting the **Only These** or the **All But These** options. Select **All Users** as long as you are not in a production process.

7.4.4.2 Start the dmeexec Server

At the end of the installation, PDM proposes to start *DMEExec* for you. You can modify the service options in Windows NT to allow an automatic start. You can also stop and restart the *PDMExec* service, using the scrolling Service menu on the PDM graphical Interface.

7.4.4.3 Windows NT Files

After the software is extracted and the installation program has completed, the following files and directories exist:

File or Directory	Description
\Program Files\PDM	Base directory for all installed PDM files.
\Program Files\PDM\pdmcfg.exe	Configuration utility.
\Program Files\PDM\README	File containing installation information and known problems

7.4.4.4 Additional Notes

PDM software installation automatically configures the DMERoute service to start automatically at system startup. You can also control the start and stop of this service using the pdmcfg.exe utility or the Services control panel.

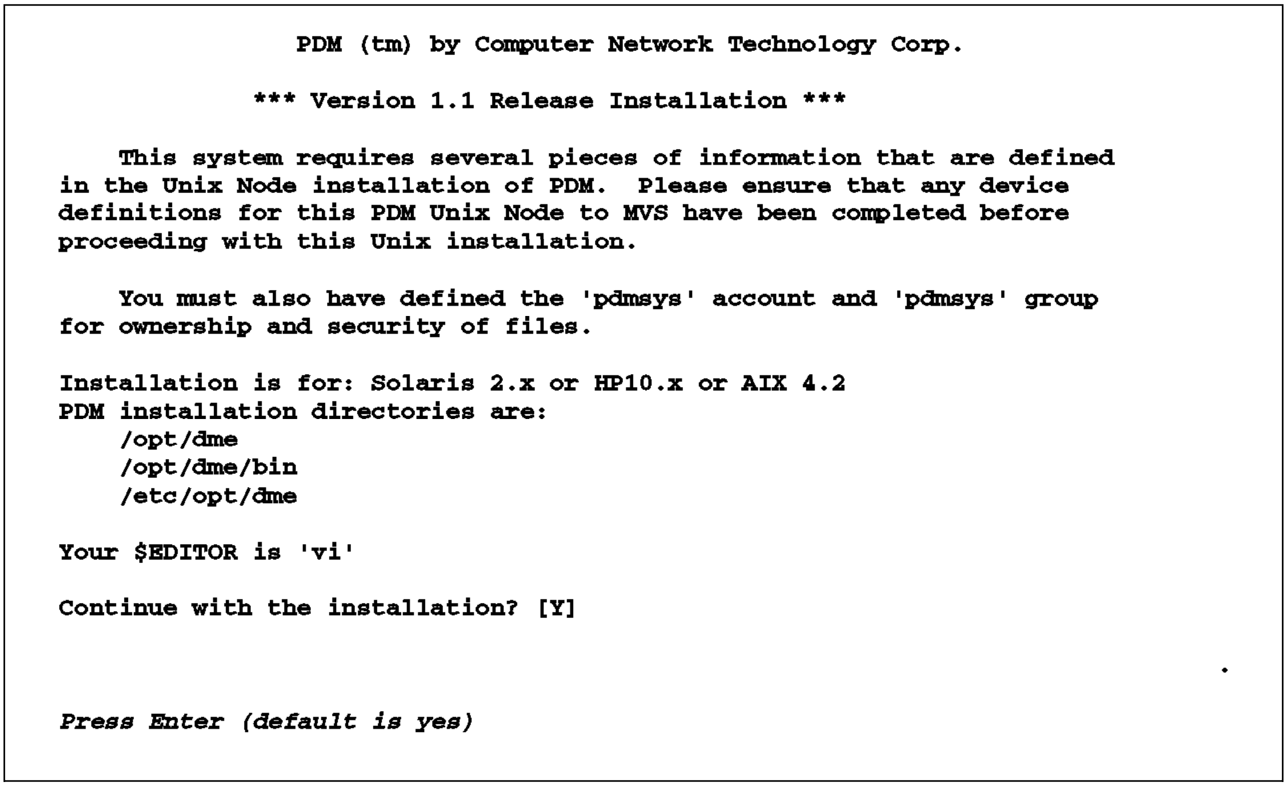
Chapter 8. Install and Configure PDM Network-Attached or Downstream Nodes

This chapter explains how to install PDM network-attached nodes (also called downstream nodes) on UNIX or Windows NT platforms.

8.1 UNIX Downstream Nodes

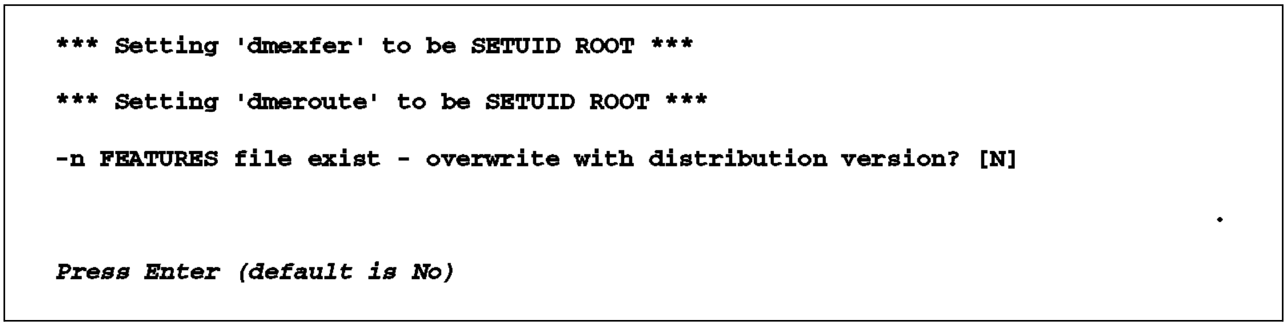
For the installation part of the IBM exPress Data Mover software on UNIX nodes, please refer to Chapter 5, “Install and Configure IBM InfoSpeed” on page 31 and check for your specific operating system. The machine to be set up as a downstream node is named *lutetia*, that will use *pisces* as its gateway node.

After the *install_pdm* program is started, a sequence of prompts appear. The first is shown in Figure 78, the second in Figure 79 on page 98 and the third and fourth in Figure 80 on page 98 and Figure 81 on page 98.



```
                PDM (tm) by Computer Network Technology Corp.

             *** Version 1.1 Release Installation ***

    This system requires several pieces of information that are defined
in the Unix Node installation of PDM.  Please ensure that any device
definitions for this PDM Unix Node to MVS have been completed before
proceeding with this Unix installation.

    You must also have defined the 'pdmsys' account and 'pdmsys' group
for ownership and security of files.

Installation is for: Solaris 2.x or HP10.x or AIX 4.2
PDM installation directories are:
    /opt/dme
    /opt/dme/bin
    /etc/opt/dme

Your $EDITOR is 'vi'

Continue with the installation? [Y]


Press Enter (default is yes)
```

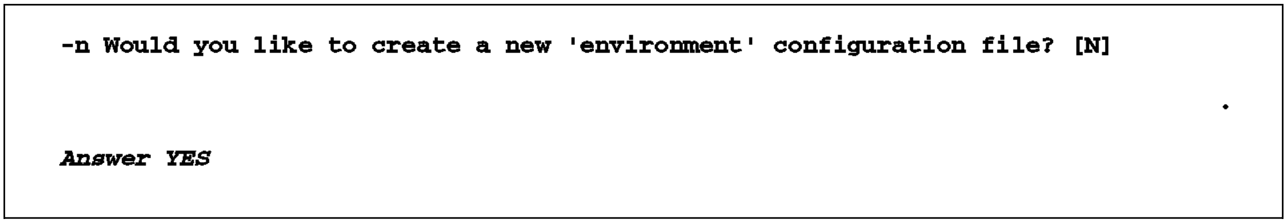
Figure 78. PDM Configuration Menu on an UNIX Downstream Node (1)



```
*** Setting 'dmexfer' to be SETUID ROOT ***

*** Setting 'dmeroute' to be SETUID ROOT ***

-n FEATURES file exist - overwrite with distribution version? [N]
```

Press Enter (default is No)

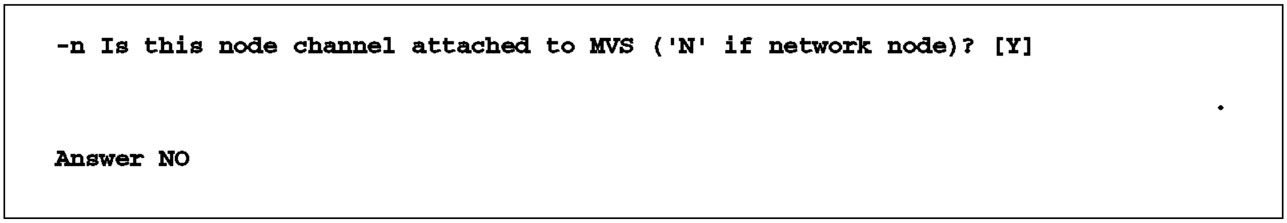
Figure 79. PDM Configuration Menu on an UNIX Downstream Node (2)



```
-n Would you like to create a new 'environment' configuration file? [N]
```

Answer YES

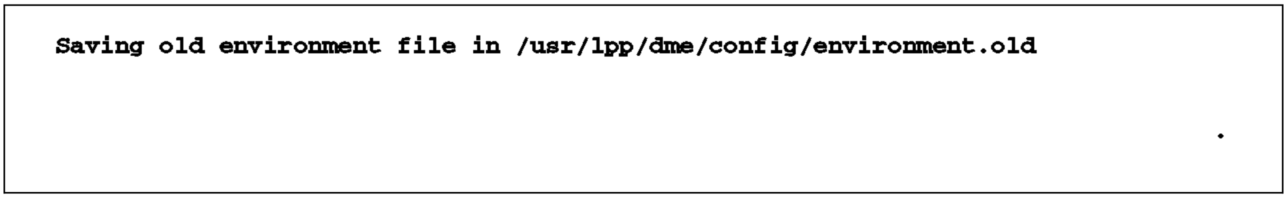
Figure 80. PDM Configuration Menu on an UNIX Downstream Node (3)



```
-n Is this node channel attached to MVS ('N' if network node)? [Y]
```

Answer NO

Figure 81. PDM Configuration Menu on an UNIX Downstream Node (4)



```
Saving old environment file in /usr/lpp/dme/config/environment.old
```

Figure 82. PDM Configuration File Complete for an UNIX Downstream Node

Then the configuration file is edited as follows, and you need to modify some parameters. The parameters that need to be modified are indicated with arrows.

```
########################################################
# @(#)env.head  7.1 24 Jun 1997
# Environment Values for PDM
# (You will probably need to edit lines with ’***’)
########################################################
# Basic Parameters that you must set or change
########################################################
### Debug level (0-9) - Set to 5 until working then drop to 0
to 0.
DebugLevel=5
### The 4 char SMF id of the MVS system.
TargetHost=3090   <-------------(1)
```

```
### For PDM network-attached node, uncomment and provide name of the PDM node router
SessionHost=pisces  <-----------(2)

### See installation manual - Communication route should be first
two subchannels
#Route=3090,0:0     <-----------(3) #
# See installation manual - Communication route should be first two
#subchannels
#..............
# Remaining subchannels are for data transfer
#..............

#MemBuffers=4        <-----------(4)

######################################################################
#
# Basic Parameters that you may optionally change
#
######################################################################
#..............
######################################################################
#
# Advanced Parameters *** DO NOT CHANGE WITHOUT UNDERSTANDING THEM ***
#
######################################################################

# Define the hostname that we should use for system with multiple
# network interfaces.  Set this to the fastest "network" name if not
# the same as "uname -n"

# ............

# Task definitions: <task name>,<task path>  - DO NOT CHANGE
Task=DMES,/usr/dme/bin/dmexfer
Task=DMEGW,/usr/dme/bin/dmeserv
```

Notes:

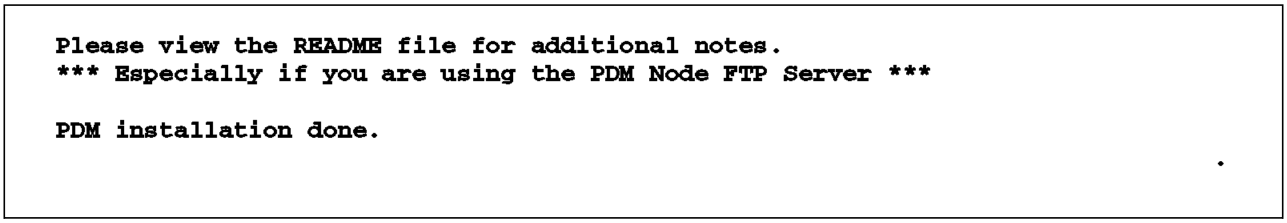
(1) As for channel-attached nodes, the SMF-ID of your default OS/390 system should be specified.

(2) *SessionHost* is the name of the channel-attached node through which this node will communicate with the OS/390 system. This name is also specified as the label of the HDEF statement in OS/390 defining this node as a subhost (see 6.3.1, "The DMETABLE Program" on page 51).

(3) As this is a network-attached node, do not define either Route or Device statements.

(4) The *MemBuffers* statement should be commented for network-attached nodes.

Note: The above sample configuration file is not complete. The Advanced Parameters, such as *TapeUnits* and *User Access Restriction* need to be updated, as for channel-attached nodes.



```
Please view the README file for additional notes.
*** Especially if you are using the PDM Node FTP Server ***

PDM installation done.
```

Figure 83. PDM Installation Complete for a UNIX PDM Network-Attached Node

Then the *dmeexec* service needs to be configured and started, so that *dmeroute* can start the file process on this node.

To configure the *dmeexec* server:

- Edit the */etc/services* file and modify it by adding the following line:

  ```
  dmeexec  6880/tcp
  ```

- Edit the */etc/inetd.conf* and add the following line:

  ```
  dmeexec stream tcp nowait root /usr/dme/bin/dmeexec
  ```

  (*/usr/lpp/dme/bin/dmeexec* for AIX systems)

- Stop and restart *inetd*:
  - Check the *inetd* process ID (pid) by typing

    ```
    ps -ef | grep inetd
    ```

  - Make *inetd* reread its configuration file by typing

    ```
    kill -HUP <pid>
    ```

After you install the IBM exPress Data Mover, add this entry to the /etc/inittab file:

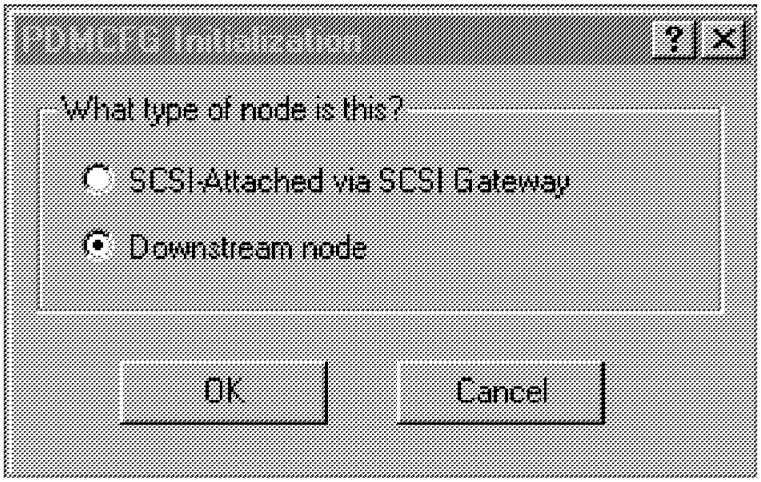
```
pdm:2:once/usr/lpp/pdm/bin/start_pdm
```

This allows dmeexec to start automatically at system startup.

8.2 Windows NT Downstream Nodes

For the installation part of the IBM exPress Data Mover software on UNIX nodes, please refer to 7.4, “Windows NT Installation Planning” on page 87.

After the *pdmcfg.exe* program has started, the prompt screen shown in Figure 84 on page 101 appears.



PDMCFG Initialization

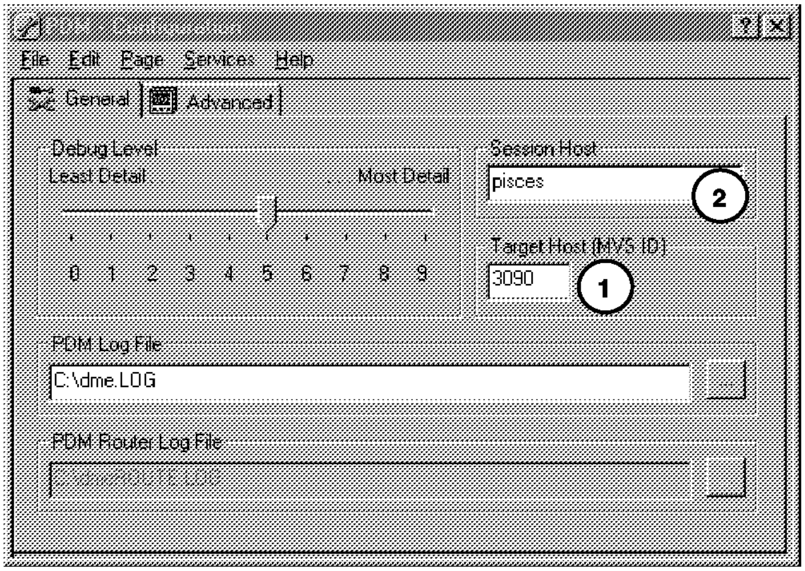
What type of node is this?

○ SCSI-Attached via SCSI Gateway

● Downstream node

Figure 84. PDM Configuration Menu on Windows NT Downstream Node (1)

In Figure 84, select **Downstream node**.. Figure 85 and Figure 86 on page 102, define the environment for IBM exPress Data Mover running under Windows NT as a network-attached or downstream node. The information to be specified is similar to the information specified in the configuration file used in a network-attached UNIX system; see the sample configuration file in 8.1, “UNIX Downstream Nodes” on page 97.



Debug Level: Least Detail 0 1 2 3 4 5 6 7 8 9 Most Detail (set to 5)

Session Host: pisces

Target Host (MVS ID): 3090

PDM Log File: C:\dme.LOG

PDM Router Log File:

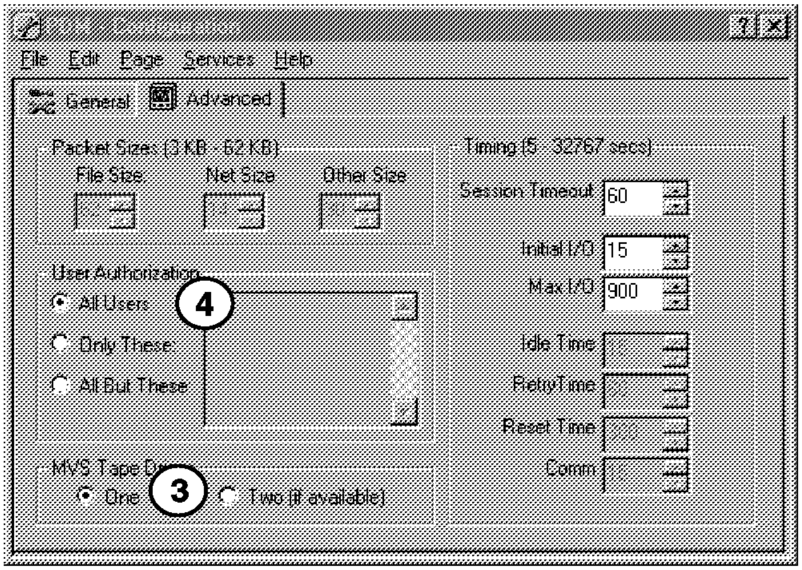
Figure 85. PDM Configuration Menu on Windows NT Downstream Node (2)

Notes:

(1) In Figure 85, in folder *General*, the SMFID of your default OS/390 should be specified.

(2) On this second panel, you should also specify the Session Host by entering the name of the channel-attached node through which this node will communicate with the OS/390 system. This name is also specified as the label of the HDEF statement in OS/390 defining this node as a subhost, see 6.3.1, “The DMETABLE Program” on page 51.

The recommendation for the Debug Level, shown in Figure 85 on page 101, is to set it to 5 until the system is working and then drop to 0.



Session Timeout 60

Initial I/O 15

Max I/O 900

● All Users
○ Only These
○ All But These

● One ○ Two (if available)

Figure 86. PDM Configuration Menu on Windows NT Downstream Node (3)

On the second panel, Figure 86, in folder *Advanced*, you should select the following:

(3) The number of tape drives that you can allocate simultaneously in MVS. Select the appropriate **MVS Tape Drives** option, if you intend to send data from the NT server directly to tapes on MVS.

(4) Specify which users can access the IBM exPress Data Mover. Any user can access the IBM exPress Data Mover, unless you specify otherwise by selecting the **Only These** or the **All But These** options. We recommend selecting **All Users** as long as you are not in a production process.

After you install the IBM exPress Data Mover, the PDM configuration program proposes to start dmeexec. Answer **yes**. You can now use the dmeftp interface as on a PDM channel-attached node

Chapter 9. Configure a PDM FTP Server for Non-PDM Nodes

This chapter describes how the PDM FTP server is configured.

9.1 PDM FTP Server on UNIX

If you are planning to use non-PDM clients, you need to configure a channel-attached node as a PDM FTP server and to start a specific daemon. There is no additional configuration of the PDM system itself, but it is necessary to make some changes to *inetd* configuration.

For the installation part of the IBM exPress Data Mover software on UNIX nodes, please refer to Chapter 5, "Install and Configure IBM InfoSpeed" on page 31 and check for your specific operating system.

9.1.1 Connection to One OS/390 Host

The steps are:

1. Establish a unique pair of ports to use (traditionally ports number 58 and 59). To reserve this port pair, change the NIS-services-table (*/etc/services*) by adding the following entry:

   ```
   dmeftp 59/tcp
   ```

   Traditionally, port 59 is used for any private file service, but any free port number can be chosen. However, be aware that port 59 is the control path, and UNIX assumes that port 58 is also free for data transfers. FTP operations always require a pair of paths, only the control port needs to be specified.

2. Edit the */etc/inetd.conf* file to let the INET services daemon *inetd* know about the new server by adding the following:

   ```
   dmeftp stream tcp nowait root /usr/dme/bin/dmeftp in.dmeftp -server
   ```

   (*/usr/lpp/dme/bin/dmeftp* for AIX systems)

3. Stop and restart the *inetd* daemon:

   - Check the *inetd* process ID by typing

     ```
     ps -ef | grep inetd
     ```

   - Make *inetd* reread its configuration file by typing

     ```
     kill -HUP <pid>
     ```

4. Verify the definitions by connecting to the defined PDM FTP server from the non-PDM. Type the following:

   ```
   $ftp PDM_FTP_server 59
   ```

   The system will answer:

   220 PDM FTP server V1.0 ready

   Name (PDM_FTP_server:user) Type your MVS userid.

9.1.2 Connection to Multiple OS/390 Servers

The PDM FTP server can be configured to handle data connections to multiple MVS PDM servers. Each server requires a specific pair of port numbers. To provide them, do the following:

1. Repeat the PDM port definition in the */etc/services/* file described above, such as:

   ```
   dmeftp1  59/tcp
   dmeftp2  103/tcp
   ```

   assuming that port 102 and port 103 are free.

2. Edit the */etc/inetd.conf* file to let the INET services daemon *inetd* know about the new server by adding the -open <smf-id> parameter:

   ```
   dmeftp1 stream tcp nowait root /usr/dme/bin/dmeftp in.dmeftp
   -server -open MVS1
   dmeftp2 stream tcp nowait root /usr/dme/bin/dmeftp in.dmeftp
   -server -open MVS2
   ```

3. Stop and restart the *inetd* daemon :

   - Check the *inetd* process ID by typing

     ```
     ps -ef | grep inetd
     ```

   - Make *inetd* reread its configuration file by typing

     ```
     kill -HUP <pid>
     ```

9.2 PDM FTP Server on Windows NT

For the installation part of the IBM exPress Data Mover software on Windows NT nodes, please refer to 7.4, “Windows NT Installation Planning” on page 87.

To be a PDM FTP server, the Windows NT system must first be configured as a standard TCP/IP FTP server. Then, PDM selects automatically ports 58 and 59, and adds a DMEFTP service, with automatic startup. If more than one PDM MVS server is to be defined, then the NT system administrator must define specific DMEFTP services, as described for the UNIX environment. One DMEFTP service is added for each defined PDM MVS server.

It is possible to edit and modify the port numbers selected by opening the **properties** field for each service.

9.3 Non-PDM Node

No change is required on the non-PDM node. Just start an FTP session using the communication port that you specified. For example:

```
ftp gemini.tucson.ibm.com 59
```

Here *gemini* is a PDM channel-attached node, which is now also used as a PDM FTP server.

Please refer to Chapter 10, “Using MVS PDM Clients” on page 105 for more explanations about using the TCP/IP FTP interface to IBM exPress Data Mover.

Chapter 10. Using MVS PDM Clients

This chapter describes how to operate and maintain the PDM MVS clients. Before using this chapter, install the PDM software and configure the PDM MVS server according to the instructions in Chapter 6, "PDM MVS Installation and Configuration" on page 43.

All examples in this chapter use the configuration shown in Figure 87.

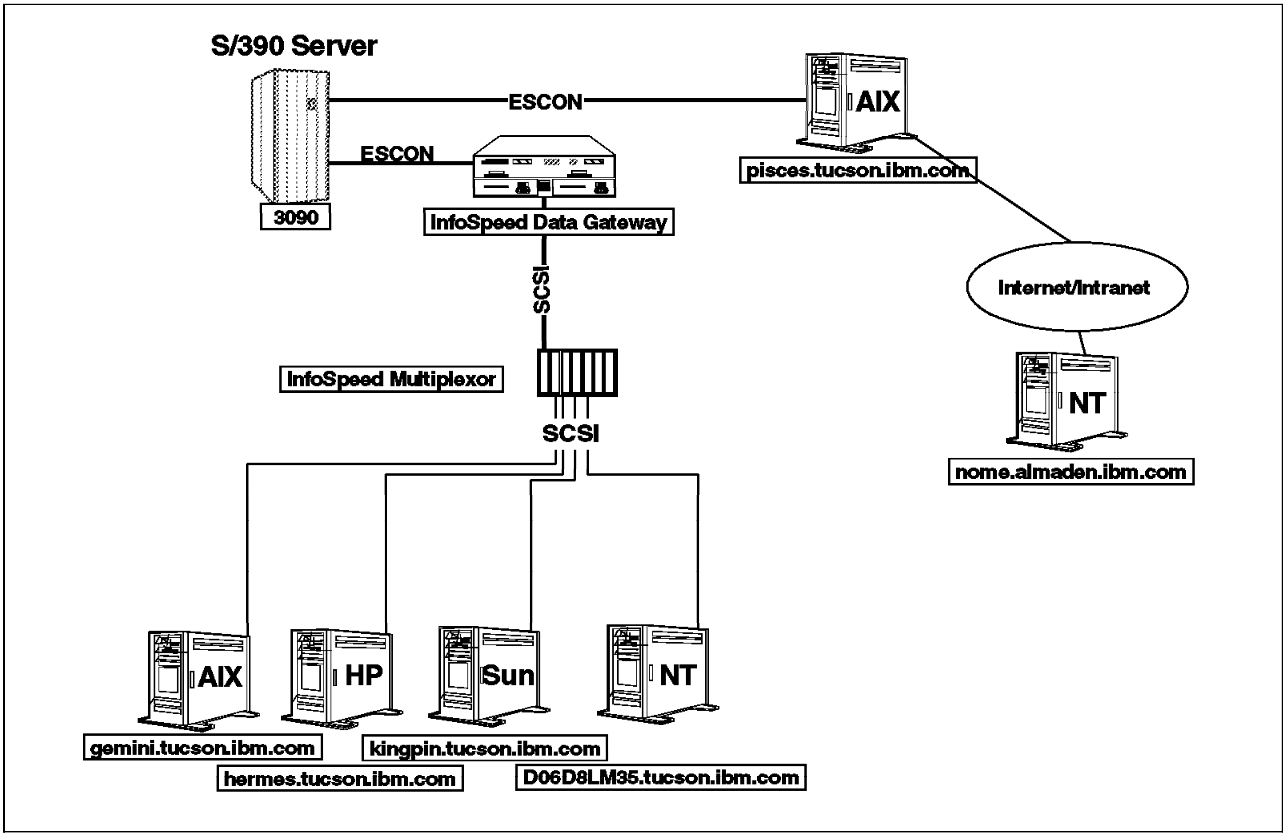


Figure 87. The Configuration Used in the MVS PDM Client Examples

10.1 PDM MVS Client

There are three PDM interfaces for the MVS user:

- **PDM MVS File Transfer Client (DMEFTP)**. This is a command language that closely resembles and is modeled after the TCP/IP FTP language. However, MVS TCP/IP is not used by the product. The program is invoked either as a batch job, or interactively under the time sharing option (TSO).
- **PDM MVS ISPF Client (DMESISPF)**. This is an interactive system productivity facility (ISPF) application with a ″fill-in-the-blank″ style panel for files to be copied. DMESISPF also has a management application called *MVS ISPF Event Manager* that provides a facility on the PDM MVS subsystem for monitoring activity, displaying transaction results, canceling transactions, and changing the state of transfer tasks.

- **PDM SubSystem Interface (SSI)**. The PDM function that supports BSAM and QSAM is called the *subsystem interface* (SSI). The SSI provides users with the ability to execute MVS applications unmodified and access data on remote nodes. Parameters are specified in JCL or through dynamic allocation parameters that describe the node, node access, and file name. The MVS application can then read and write data to and from files or executables on remote nodes.

The remote nodes (UNIX or Windows NT) are referred to by their host name, which has been defined by the MVS system programmer in the HDETABLE when the product was installed (see Chapter 6, “PDM MVS Installation and Configuration” on page 43 for more information). The MVS user needs an account and password on each remote node to access files but does not need to logon to remote nodes to move files to or from those nodes.

10.1.1 Using the PDM MVS File Transfer Client (DMEFTP)

The DMEFTP program can be invoked either as a batch job or interactively under TSO. The PDM MVS File Transfer Client provides the ability to copy ASCII/EBCDIC or binary files between the MVS host and remote nodes, make and remove directories on remote nodes, delete and rename remote files, list directories, pipe data, and perform other remote operations on the local MVS host.

10.1.1.1 Executing DMEFTP in Batch Mode

In batch mode, Job Control Language (JCL) is used to pass parameters on the EXEC statement and to provide the DMEFTP input control statements to the program. The required JCL statements are:

```
//MYJOB JOB (USER,XXXX),MSGLEVEL=1
//S1      EXEC PGM=DMEFTP,PARM=(value,value,value)
//STEPLIB  DD  DSN=hlq.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1 DD  *
(PDM Control Statements)
/*
```

The following are the allowable execution PARM values on the EXEC statement:

SERIAL Serialize all transactions

or

PARALLEL Run transactions in parallel

SYNC Synchronize (wait for completion)

or

NOSYNC Do not synchronize

SOURCE Specify command library member

For an overview of the DMEFTP command language and summary of commands, refer to *PDM User's Guide and Reference*, GA32-0388.

10.1.1.2 Copying a UNIX file to MVS (GET)

The example in Figure 88 shows JCL to copy a remote node ASCII file to a new MVS sequential data set using the batch interface. Comments prefixed with an asterisk (*) describe each statement. Code UNIX and Windows NT parameters, such as file names and host names, as they are coded on the hosts because these are case sensitive. MVS parameters are case insensitive and need not be coded in upper case.

```
//MYJOB JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP
//STEPLIB  DD  DSN=hlq.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1 DD  *
* Open remote node named ″gemini″
   open gemini
* Turn off echoing so password does not appear in output
   echo off
* Identify user ID and password to ″gemini″
   log user01 mypasswd
* Turn echoing back on to display commands
   echo on
* Specify MVS new data set parameters
   lsite lrecl 80
   lsite blksize 800
   lsite recfm fb
   lsite tracks
   lsite primary 5
   lsite unit sysda
* Specify TSO-user ID to be notified when transfer completed
   notify mymvsid
* Set remote current working directory
   cd /home/user01
   ASCII
* Copy file ″/home/user01/myfile.txt″ to new MVS data set ″myfile.user.text″
   get myfile.txt myfile.user.text
* Close the connection to ″gemini″
   close
* End the job
   quit
/*
```

Figure 88. Sample JCL for PDM MVS File Transfer Client GET Command

This example creates a new data set *myfile.user.text* on MVS. New data sets created by PDM are automatically cataloged in MVS if the copy function completes successfully. If it does not, they are deleted unless the ″CHKPT ON″ parameter has been coded. On installations that do not run Data Facility Storage Management Subsystem (DFSMS), the user can specify the parameters or information of the new data set with the *lsite* control statements. If no *lsite* control statements are coded and no DFSMS parameters are used, PDM will post an error and terminate.

In a DFSMS environment, most *lsite* control statements are unnecessary for a new data set because DFSMS automatically defines the data set characteristics through the use of data classes. At least one lsite control statement is required for the data class parameter. The *lsite* statements that need to be coded will depend on the installation′s data class setup in DFSMS. Data set characteristics defined in data classes can be overridden with *lsite* control statements. Check with the installation′s storage administrator for data class setup. See Appendix D, "A Short Introduction to DFSMS" on page 201 for more information on DFSMS and data set management.

See Chapter 12, "Data Copy Sharing Samples" on page 141 for more samples using the DMEFTP client in batch mode.

10.1.1.3 Executing PDM in Interactive Mode

To run *DMEFTP* interactively, a CLIST is provided with PDM which calls the program under TSO. To invoke the program, enter DMEFTP (or whatever the system programmer has called the CLIST) on the TSO command line. The following will appear:

```
EXECUTION ENVIRONMENT: TSO,   SYNCHRONOUS,  SERIAL
EXPRESS DATA MOVER - TSO FTP PROCESSOR - V1.1
DMEFTP==> _
```

The DMEFTP interactive session has now been initiated. DMEFTP prompts the user for input with lines that begin with DMEFTP==>.

Figure 89 shows an example of a completed interactive session. The user has copied file *myfile2.txt* from directory */home/user01/tmp*, from the remote node named *gemini*, to a new MVS data set *myfile2.user.text*. Note the similarities between the commands entered in the interactive session in Figure 89 and the input commands in the JCL in Figure 88 on page 107.

```
EXECUTION ENVIRONMENT: TSO,   SYNCHRONOUS,  SERIAL
EXPRESS DATA MOVER - TSO FTP PROCESSOR - V1.1
DMEFTP==> open gemini
open gemini
DMEFTP==> log user01 mypasswd
DME552I CONNECTION TO HOST gemini         FOR USER user01       ACCEPTED
DMEFTP==> lsite lrecl 80
DMEFTP==> lsite blksize 800
DMEFTP==> lsite recfm fb
DMEFTP==> lsite tracks
DMEFTP==> lsite primary 5
DMEFTP==> lsite unit
DMEFTP==> notify user01
DMEFTP==> cd /home/user01/tmp
/home/user01/tmp
DMEFTP==> ASCII
DMEFTP==> get myfile2.txt myfile2.user.text
DMESYS     - REQ(05600398), RC=(000), FUNCTION(GET    ) COMPLETE CN(INTERNAL)
REQ(05600398) STATUS(COMPLETE) BYTES(        12,100) CC(0000) RC(0000)
DMEFTP==> close
DMEFTP==> quit
***
```

Figure 89. PDM MVS File Transfer Client Commands for an Interactive TSO Session

The interactive session begins in the SYNCHRONOUS, SERIAL mode. To change modes, simply enter the desired mode on the prompt line. For example, entering parallel will result in the following:

```
DMEFTP==> parallel
EXECUTION ENVIRONMENT: TSO,  ASYNCHRONOUS,  PARALLEL
```

The same DMEFTP commands are used in both batch and interactive modes. Refer to the *PDM User's Guide and Reference*, GA32-0388, for a detailed description of the DMEFTP commands.

10.1.1.4 Using Source Libraries

A command library script can also be used to execute DMEFTP commands at startup. To access the command library, add a DD statement to the JCL with ddname *DMECLIB* and code the data set name of the library that contains the commands. Code the *SOURCE(nnnnnnnn)* parameter on the *PARM=* keyword of the EXEC statement where *nnnnnnnn* is the member name:

```
//MYJOB   JOB ,'PDMSYS',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=H,CLASS=A
//STEP01  EXEC PGM=DMEFTP,PARM=('SOURCE(nnnnnnnn)')
//STEPLIB  DD  DSN=hlq.SDMELOAD,DISP=SHR
//DMECLIB  DD  DSN=user.SDMESRCE.CNTL,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  *
(PDM Control Statements)
/*
```

where *user*.SDMESRCE.CNTL is the data set name of the command library (you can allocate any data set name).

Figure 90 shows sample JCL to execute command library member AUTOLOG from data set *user*.SDMESRCE.CNTL.

```
//DMEGET4 JOB ,'PDMSYS',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=H,CLASS=A,NOTIFY=HSMATH1
//STEP01  EXEC PGM=DMEFTP,PARM=('SOURCE(AUTOLOG)')
//STEPLIB  DD  DSN=hlq.SDMELOAD,DISP=SHR
//DMECLIB  DD  DSN=user.SDMESRCE.CNTL,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  *
   ascii
   lsite tracks
   lsite primary 1
   lsite secondary 1
   lsite unit sysda
   lsite recfm vb
   lsite lrecl 32756
   lsite blksize 32760
   notify hsmath1
   get /tmp/hdetable.mvs hsmath1.test.pdmsrce
   close
/*
```

Figure 90. Command Library JCL Sample for AUTOLOG

The member AUTOLOG contains the following DMEFTP commands:

```
open gemini
echo off
log userid password
echo on
```

Figure 91 on page 110 shows sample JCL to execute multiple command library members:

```
//DMEGET4 JOB ,'PDMSYS',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=H,CLASS=A,NOTIFY=HSMATH1
//STEP01  EXEC PGM=DMEFTP,PARM=('SOURCE(AUTOLOG)')
//STEPLIB  DD  DSN=hlq.SDMELOAD,DISP=SHR
//DMECLIB  DD  DSN=user.SDMESRCE.CNTL,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  *
   source siteasci
   get /tmp/hdetable.mvs hsmath1.test.pdmsrce
   close
/*
```

Figure 91. Command Library JCL Sample to Execute Multiple Members

Along with the member AUTOLOG, member SITEASCI contains the following DMEFTP commands:

```
ascii
lsite tracks
lsite primary 1
lsite secondary 1
lsite unit sysda
lsite recfm vb
lsite lrecl 32756
lsite blksize 32760
notify hsmath1
```

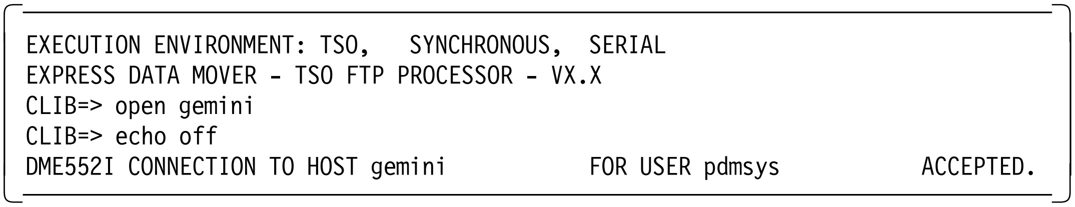
Both members are in the same command library.

If running the DMEFTP program interactively, edit the DMEFTP clist member and add the following to the PARM values on the CALL line: SOURCE(membername), where membername is the member name in the command library containing the commands. The following lines also need to be added in the CLIST:

```
FREE F(DMECLIB)
ALLOC F(DMECLIB) DSN(myLib.cntl)
```

where *mylib.cntl* is the data set that contains the command library.

Using the same command library members AUTOLOG and SITEASCI from the previous example, when the DMEFTP clist is invoked, an interactive user sees what is shown in Figure 92.



```
EXECUTION ENVIRONMENT: TSO,   SYNCHRONOUS,  SERIAL
EXPRESS DATA MOVER - TSO FTP PROCESSOR - VX.X
CLIB=> open gemini
CLIB=> echo off
DME552I CONNECTION TO HOST gemini           FOR USER pdmsys           ACCEPTED.
```

Figure 92. Using Command Library in Interactive Session

The first member, AUTOLOG, is invoked automatically with the clist, as long as SOURCE(AUTOLOG) has been coded in the PARM= keyword in the clist. To invoke the second command library member, enter **source siteasci** on the DMEFTP command line, as shown in Figure 93 on page 111.

```
DMEFTP==> source siteasci
CLIB=> ASCII
CLIB=> LSITE TRACKS
CLIB=> LSITE PRIMARY 1
CLIB=> LSITE SECONDARY 1
CLIB=> LSITE UNIT SYSDA
CLIB=> LSITE RECFM VB
CLIB=> LSITE LRECL 32756
CLIB=> LSITE BLKSIZE 32760
CLIB=> NOTIFY HSMATH1
DMEFTP==> get /tmp/hdetable.mvs hsmath1.test.pdmsrce
REQ(13400382) STATUS(COMPLETE) BYTES(        13,972) CC(0000) RC(0000)
DMEFTP==> quit
```

Figure 93. Sample Code for Using Command Library in Interactive Session (Multiple Members)

The TSO CLIST contains the following statements for this example:

```
PROC 0
ALLOC F(DMECLIB) DSN(user.SDMESRCE.CNTL) SHR
ALLOC F(DMEHELP) DSN(PDM.SDMESAMP(DMEHELP)) SHR
CALL ′hlq.SDMELOAD(DMEFTP)′ ′SYNC,TSO,SOURCE(AUTOLOG)′
FREE F(SYSOUT)
FREE F(DMECLIB)
FREE F(DMEHELP)
```

The samples in Figure 91 on page 110, Figure 92 on page 110 and Figure 93 produce the same results.

10.1.2 Using the MVS ISPF Client (DMESISPF)

The MVS ISPF Client is a form based ISPF application with a ′fill-in-the-blank′ style panel on which to specify files to be copied. DMESISPF also has a management application called the *MVS ISPF Event Manager* that provides a facility for monitoring activity, displaying transactions results, canceling transactions, and changing the state of transfer tasks.

The system administrator is expected to have installed the PDM ISPF routines and panels as an integral application available under ISPF. The actual method used to invoke the main panel is installation dependent.

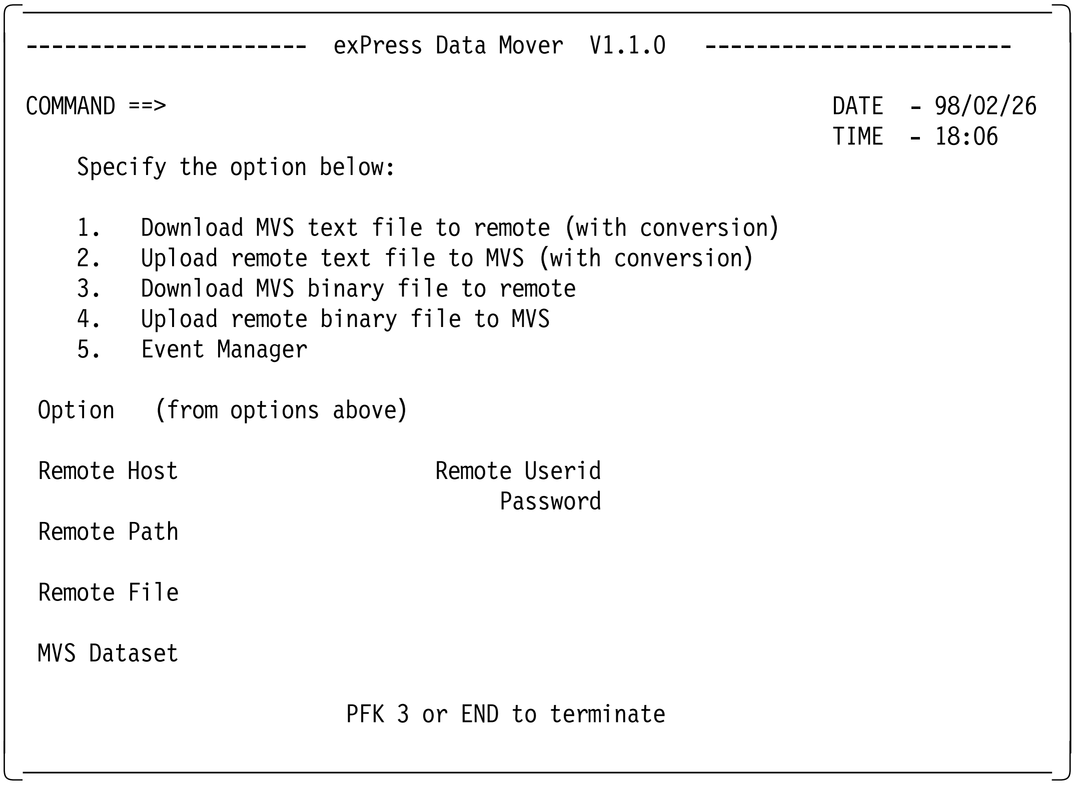
Figure 94 on page 112 shows the main MVS ISPF Client panel. Options 1 through 4 are used for file transfers and Option 5 is used to invoke the PDM Event Manager panels to display the status of requests.

The user needs to supply entries for **Remote Host**, **Remote Userid**, and **Password** for Options 1-4. The **Remote Path** is optional and can be used to specify the path. The combination of **Remote Path** and **Remote File** are used to fully qualify the remote file name. Upper and lower case entries are preserved in these fields. The **Password** field is masked and the word does not appear when entered.

The *MVS Dataset* field fully qualifies the MVS data set to be copied or written into. The system converts this data set name to upper case. The MVS data set must already exist. For PDS and GDGs, specify the member name or generation level in parentheses following the data set name. GDG references to (+1) produce an error.

Caution

New data sets can not be defined with this ISPF interface.



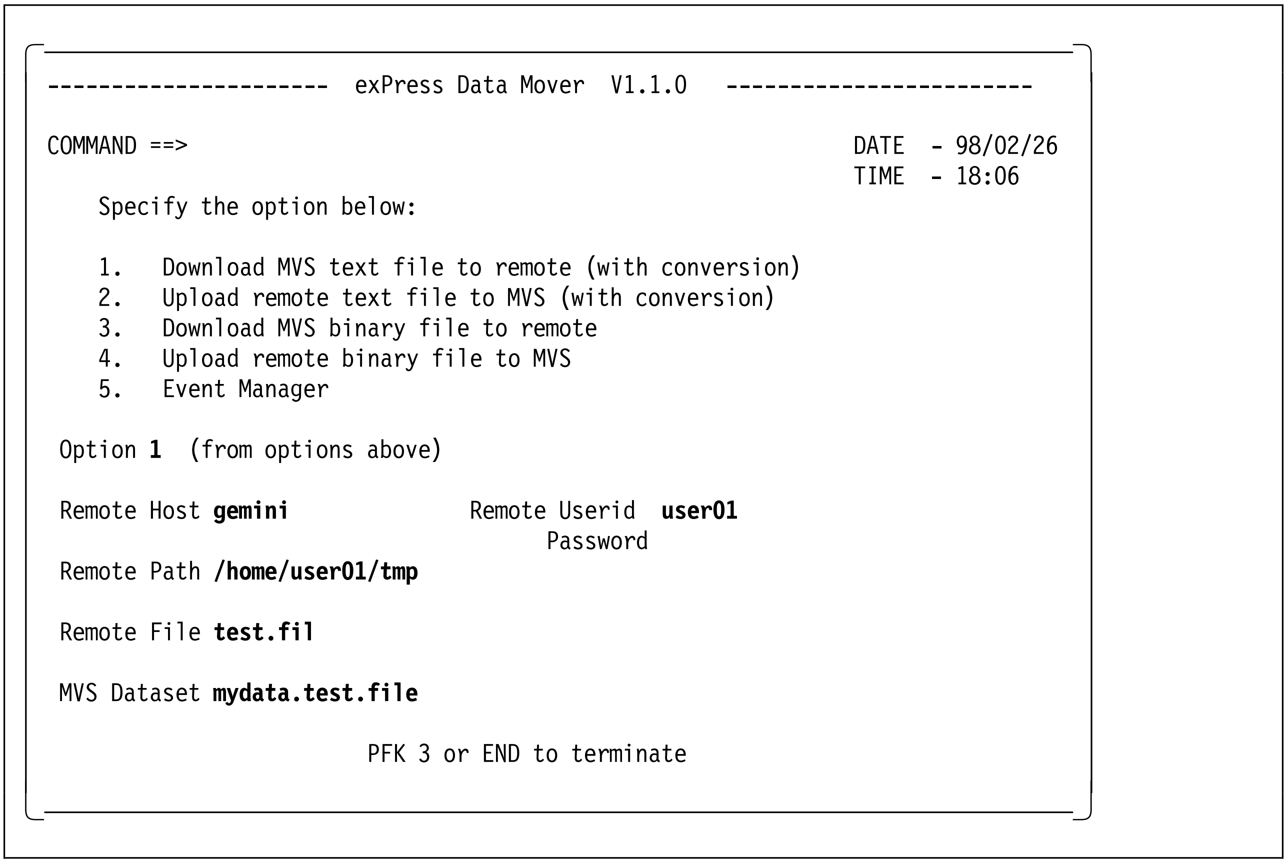
```
----------------------  exPress Data Mover  V1.1.0   ------------------------

COMMAND ==>                                                DATE  - 98/02/26
                                                           TIME  - 18:06
    Specify the option below:

    1.   Download MVS text file to remote (with conversion)
    2.   Upload remote text file to MVS (with conversion)
    3.   Download MVS binary file to remote
    4.   Upload remote binary file to MVS
    5.   Event Manager

 Option   (from options above)

 Remote Host                  Remote Userid
                                   Password
 Remote Path

 Remote File

 MVS Dataset

                       PFK 3 or END to terminate
```

Figure 94. PDM MVS File Transfer Client Main Panel

The COMMAND line is used for standard ISPF options and options for PDM are coded in the **Option** field. The cursor automatically positions at the **Option** field when the panel is invoked.

10.1.2.1 Using the MVS ISPF Client to Transfer Files

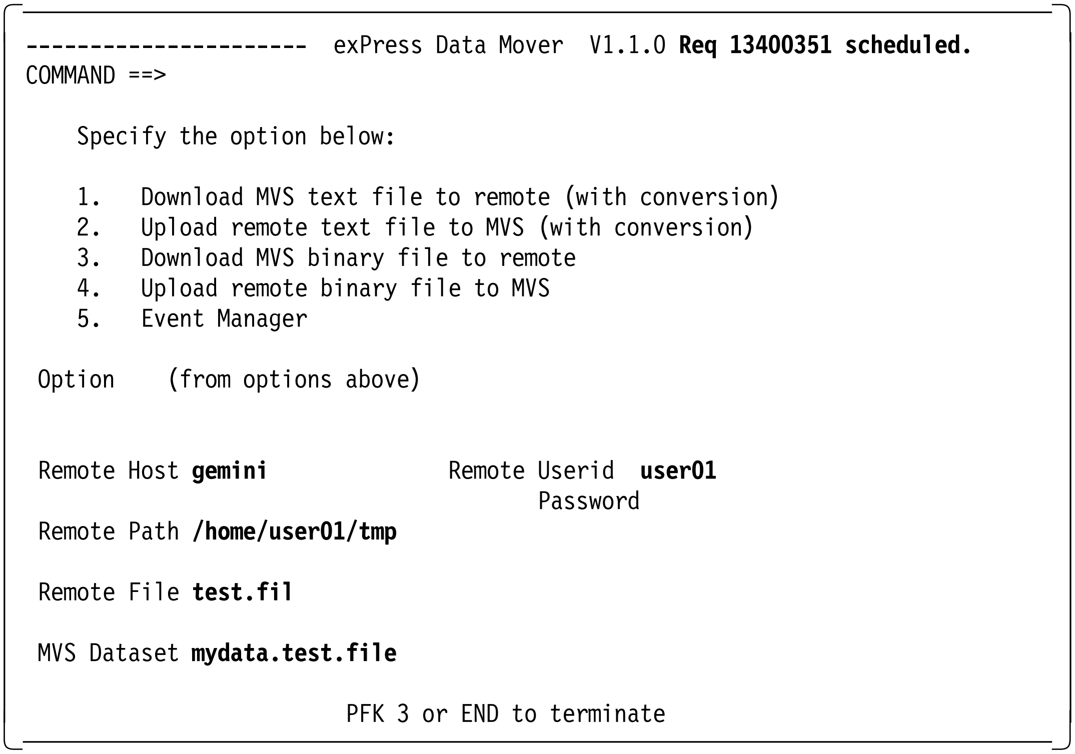
Figure 95 on page 113 is an example of the code for sending a file from MVS to a remote host.



```
----------------------  exPress Data Mover  V1.1.0   ------------------------

COMMAND ==>                                                DATE  - 98/02/26
                                                           TIME  - 18:06
    Specify the option below:

    1.   Download MVS text file to remote (with conversion)
    2.   Upload remote text file to MVS (with conversion)
    3.   Download MVS binary file to remote
    4.   Upload remote binary file to MVS
    5.   Event Manager

 Option 1  (from options above)

 Remote Host gemini             Remote Userid  user01
                                      Password
 Remote Path /home/user01/tmp

 Remote File test.fil

 MVS Dataset mydata.test.file

                         PFK 3 or END to terminate
```

Figure 95. PDM MVS File Transfer Client Panel, Download Example

After all fields have been entered, press the Enter key to schedule the transfer request. PDM responds with a transaction number if all the fields are correctly entered, or with an error message if any of the field is entered incorrectly. All responses appear in the top right-hand corner of the panel, as shown in Figure 96 on page 114.



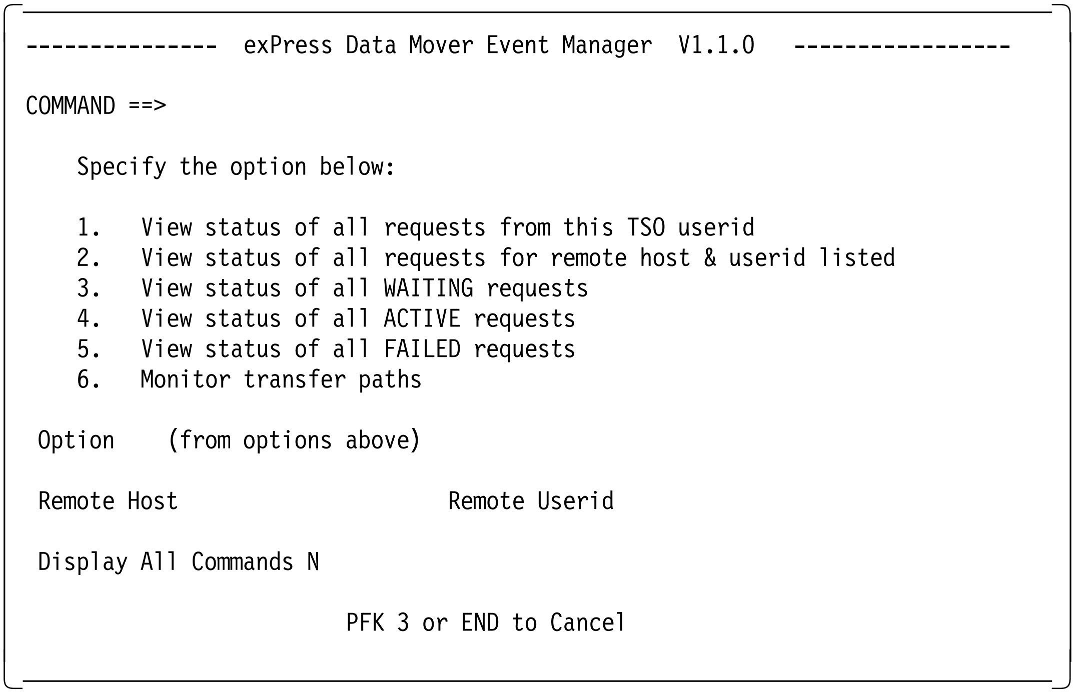
```
----------------------  exPress Data Mover  V1.1.0 Req 13400351 scheduled.
COMMAND ==>

    Specify the option below:

    1.   Download MVS text file to remote (with conversion)
    2.   Upload remote text file to MVS (with conversion)
    3.   Download MVS binary file to remote
    4.   Upload remote binary file to MVS
    5.   Event Manager

 Option    (from options above)


 Remote Host gemini              Remote Userid  user01
                                        Password
 Remote Path /home/user01/tmp

 Remote File test.fil

 MVS Dataset mydata.test.file

                         PFK 3 or END to terminate
```

Figure 96. PDM MVS File Transfer Client Panel, Download Response Example

10.1.2.2 Using the MVS ISPF Client Event Manager

The Event Manager application of the MVS ISPF Client is used to monitor all PDM transfer requests and transfer paths. Requests can be canceled and transfer paths stopped and started from the Event Manager panels.

If you select Option 5 from the MVS ISPF Client main panel, the Event Manager panel is presented, as in Figure 97 on page 115:



```
---------------  exPress Data Mover Event Manager  V1.1.0   -----------------

COMMAND ==>

    Specify the option below:

    1.   View status of all requests from this TSO userid
    2.   View status of all requests for remote host & userid listed
    3.   View status of all WAITING requests
    4.   View status of all ACTIVE requests
    5.   View status of all FAILED requests
    6.   Monitor transfer paths

 Option    (from options above)

 Remote Host                     Remote Userid

 Display All Commands N

                        PFK 3 or END to Cancel
```

Figure 97. PDM MVS File Transfer Client Event Manager Panel

As with the other panels, the COMMAND line is used for standard ISPF options and the cursor is at the Option field when invoked. Table 8 shows each option and the results of the selection.

Table 8. PDM Event Manager Panel Options

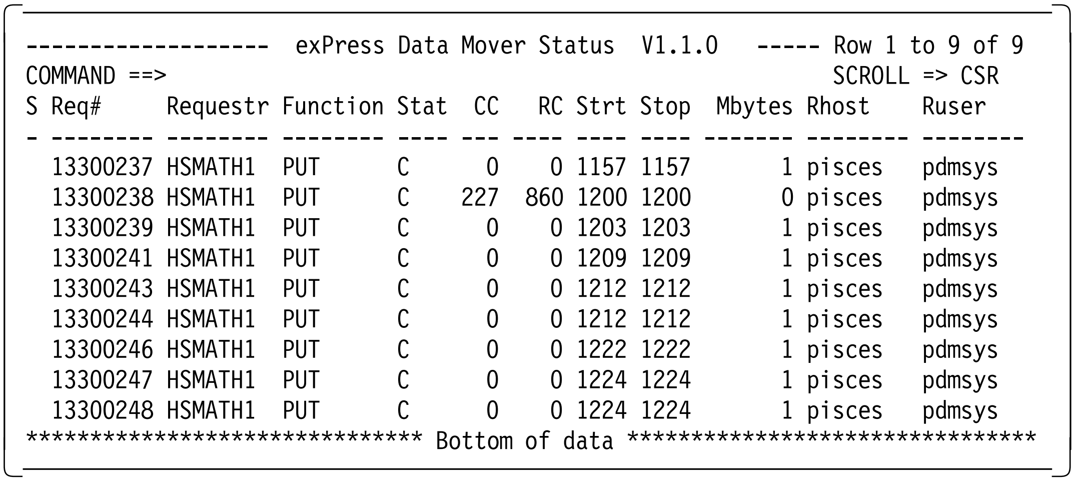
Option	Results
1	Displays the status of all requests scheduled by this TSO user. To appear on the list, the request must have been scheduled through the MVS ISPF client.
2	Displays the status of all requests scheduled for the specified remote node and user ID. The REMOTE HOST and REMOTE USERID fields are optional. If these fields are blank, **all** is assumed.
3	Displays the status of all pending or queued requests not yet executed. REMOTE HOST and REMOTE USERID are ignored.
4	Displays the status of all active requests. REMOTE HOST and REMOTE USERID are ignored.
5	Displays the status of all failed requests. REMOTE HOST and REMOTE USERID are ignored.
6	Selects the File Transfer Task Monitor application. REMOTE HOST and REMOTE USERID are ignored.

Notes:

1. Option 2 allows you to supply a partial or complete field value to limit your scan. Wildcards are allowed using an asterisk (*). The combination of characters and wildcards serves as a filter for table lookup. For example, a Remote Host name ***xyz* and Remote Userid *abc* locates all host names with characters *xyz* in positions three through five with user ID beginning with *abc*.

2. The field **Display All Commands** defaults to *N*, which prevents commands not related to data transfer, such as list directories and logons, from appearing. To display all of the commands, type *Y* in this field or delete the N and leave this field blank.

The results from selecting Options 1-5 depend on which option is chosen. Figure 98 is an example of choosing option **1**:



```
-------------------  exPress Data Mover Status  V1.1.0   ----- Row 1 to 9 of 9
COMMAND ==>                                                  SCROLL => CSR
S Req#     Requestr Function Stat  CC   RC Strt Stop  Mbytes Rhost    Ruser
- -------- -------- -------- ---- --- ---- ---- ---- ------- -------- --------
  13300237 HSMATH1  PUT      C      0    0 1157 1157       1 pisces   pdmsys
  13300238 HSMATH1  PUT      C    227  860 1200 1200       0 pisces   pdmsys
  13300239 HSMATH1  PUT      C      0    0 1203 1203       1 pisces   pdmsys
  13300241 HSMATH1  PUT      C      0    0 1209 1209       1 pisces   pdmsys
  13300243 HSMATH1  PUT      C      0    0 1212 1212       1 pisces   pdmsys
  13300244 HSMATH1  PUT      C      0    0 1212 1212       1 pisces   pdmsys
  13300246 HSMATH1  PUT      C      0    0 1222 1222       1 pisces   pdmsys
  13300247 HSMATH1  PUT      C      0    0 1224 1224       1 pisces   pdmsys
  13300248 HSMATH1  PUT      C      0    0 1224 1224       1 pisces   pdmsys
****************************** Bottom of data *******************************
```

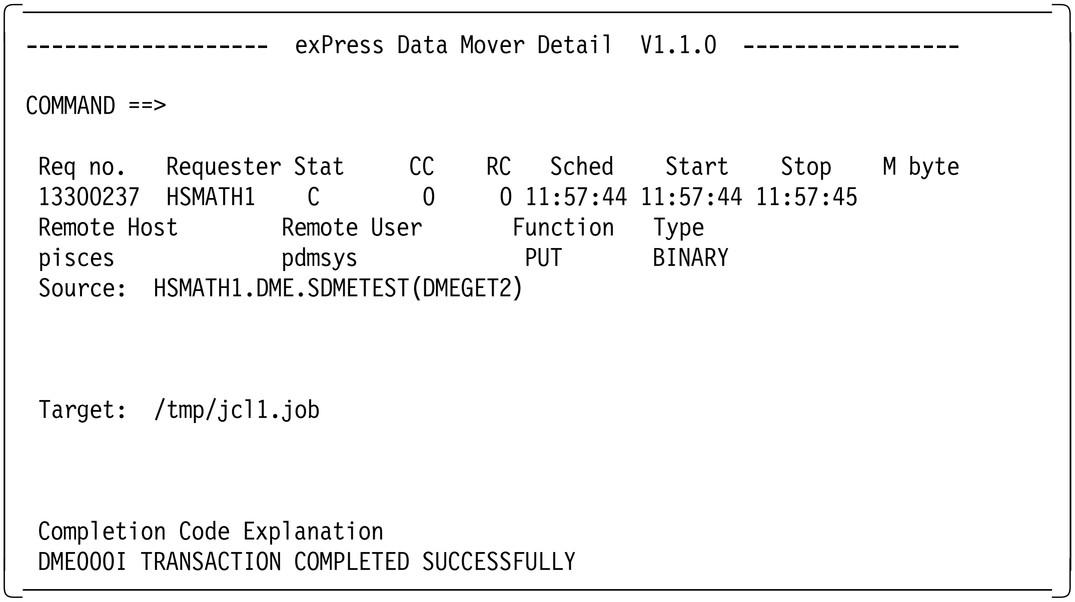
Figure 98. Event Manager Status Panel

Information on this panel is static and is not automatically updated. To obtain the current status of a particular entry, type U in the **S** field of the entry. The system refreshes only that entry with new information, including status and bytes transferred. To update multiple entries, type an U on each line to be updated. Table 9 on page 117 contains a description of each field.

Table 9. PDM Event Manager Status Panel Fields

Field	Description
S	Selection. This field is unprotected. By coding a U, D, or C in this field, you are presented with a detail screen for this entry, which displays the names of the files involved in the transfer and the type of file transfer. **U** Provides the current status of a particular entry or multiple entries. **D** Provides details about a transaction on the Event Manager Detail panel. **C** Provides a confirm cancel panel Entering S displays the explaination of the error code on the detail screen. The detail screen is shown in Figure 99 on page 118.
Req#	The request number which was presented to the user at the time the request was scheduled.
Requestr	This is the ID of the requestor that scheduled the request. It is either the TSO logon ID or the name of the job that scheduled the request.
Function	This is the function that was scheduled (GET, PUT, MKDIR for example).
Stat	The status of the request: C=completed W=wait (pending), A=active
CC	Completion code. This is valid for completed transactions only. Refer to the *PDM Codes and Messages* section of the *PDM User's Guide and Reference*, GA32-0388, for an explanation of the completion-codes.
RC	Reason code. This is valid for completed transactions only. Combined with the completion code, it provides additional information on errors. Refer to the *PDM Codes and Messages* section of the *PDM User's Guide and Reference*, GA32-0388, for an explanation of the reason codes.
Strt	Start time of the transaction in HHMM.
Stop	Stop time of the transaction in HHMM.
Mbytes	The number of megabytes transferred (rounded up).
Rhost	Remote host name.
Ruser	Remote user ID.

Event Manager Detail Panel. Entering a D or S in the selection field of the transaction provides more detail of the transaction. The panel in Figure 99 on page 118 is displayed.

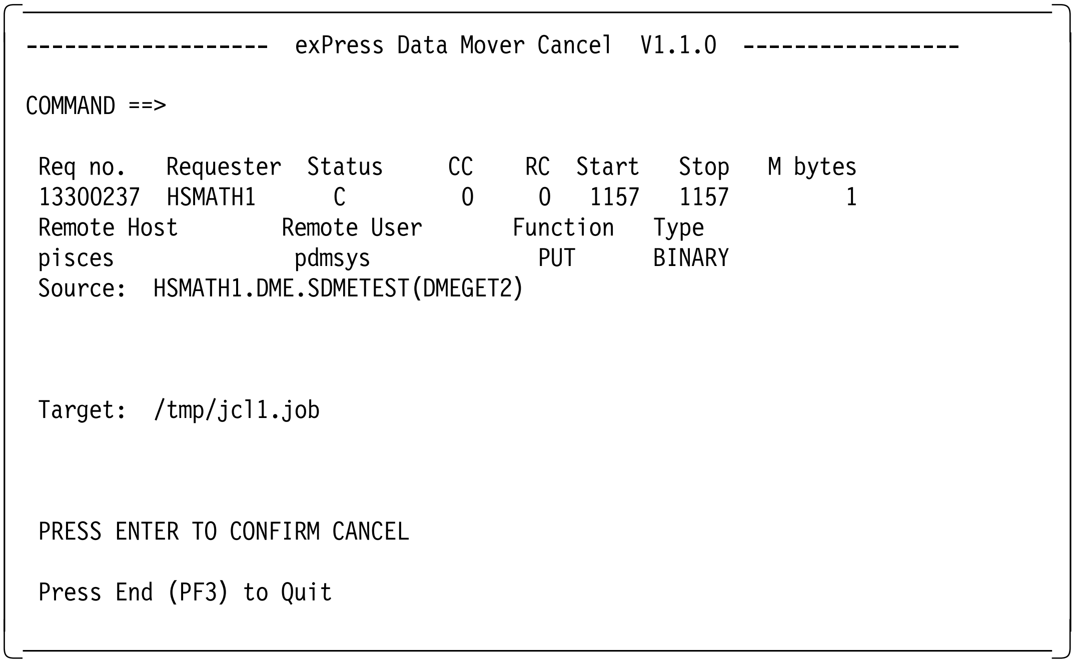


```
-------------------  exPress Data Mover Detail  V1.1.0  -----------------

COMMAND ==>

 Req no.   Requester Stat     CC    RC   Sched    Start    Stop    M byte
 13300237  HSMATH1    C        0     0 11:57:44 11:57:44 11:57:45
 Remote Host        Remote User       Function   Type
 pisces             pdmsys             PUT       BINARY
 Source:  HSMATH1.DME.SDMETEST(DMEGET2)



 Target:  /tmp/jcl1.job



 Completion Code Explanation
 DME000I TRANSACTION COMPLETED SUCCESSFULLY
```

Figure 99. Event Manager Detail Panel

Pressing Enter to return to the prior screen (Event Manager Status Panel). Use **PF3** to go back to the Event Manager Selection Panel.

Event Manager Cancel Panel Entering C in the selection field of the transaction on the Event Manager Status panel displays the Confirm Cancel panel in Figure 100.



```
-------------------  exPress Data Mover Cancel  V1.1.0  -----------------

COMMAND ==>

 Req no.   Requester  Status     CC    RC  Start   Stop   M bytes
 13300237  HSMATH1      C         0     0   1157   1157         1
 Remote Host        Remote User       Function   Type
 pisces              pdmsys             PUT      BINARY
 Source:  HSMATH1.DME.SDMETEST(DMEGET2)



 Target:  /tmp/jcl1.job



 PRESS ENTER TO CONFIRM CANCEL

 Press End (PF3) to Quit
```

Figure 100. Event Manager Cancel Panel

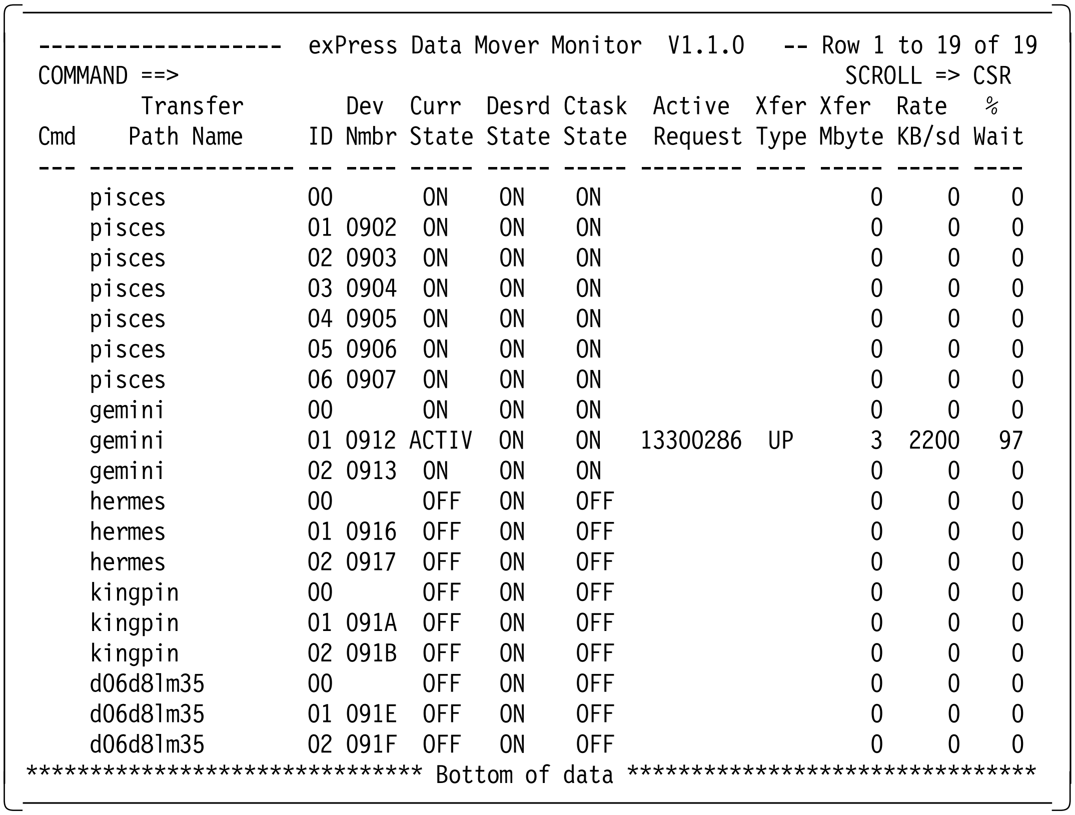
Pressing the Enter key attempts a cancel and you are immediately presented with the results. The completion of the transaction is shown by the code *222*, but it may not complete immediately as transactions in progress may have to reach a checkpoint to complete the cancel operation.

You must be the owner of the transaction (MVS userID associated with the transaction) or a privileged user as defined in the HDETABLE to cancel the request. If neither is true, the cancel request is denied.

Press the PF3 key aborts the cancel operation and returns you to the Event Manager Status Panel.

10.1.2.3 Using MVS ISPF Client to Monitor Transfer Paths

Enter 6 in the **Option** field of the Event Manager panel, Figure 97 on page 115, to display the Transfer Paths Monitor panel, Figure 101.



```
 -------------------  exPress Data Mover Monitor  V1.1.0   -- Row 1 to 19 of 19
 COMMAND ==>                                                  SCROLL => CSR
         Transfer        Dev  Curr  Desrd Ctask  Active  Xfer Xfer  Rate   %
 Cmd    Path Name     ID Nmbr State State State  Request Type Mbyte KB/sd Wait
 --- ---------------- -- ---- ----- ----- ----- -------- ---- ----- ----- ----
     pisces           00       ON    ON    ON                       0     0    0
     pisces           01 0902  ON    ON    ON                       0     0    0
     pisces           02 0903  ON    ON    ON                       0     0    0
     pisces           03 0904  ON    ON    ON                       0     0    0
     pisces           04 0905  ON    ON    ON                       0     0    0
     pisces           05 0906  ON    ON    ON                       0     0    0
     pisces           06 0907  ON    ON    ON                       0     0    0
     gemini           00       ON    ON    ON                       0     0    0
     gemini           01 0912 ACTIV  ON    ON   13300286  UP        3  2200   97
     gemini           02 0913  ON    ON    ON                       0     0    0
     hermes           00       OFF   ON    OFF                      0     0    0
     hermes           01 0916  OFF   ON    OFF                      0     0    0
     hermes           02 0917  OFF   ON    OFF                      0     0    0
     kingpin          00       OFF   ON    OFF                      0     0    0
     kingpin          01 091A  OFF   ON    OFF                      0     0    0
     kingpin          02 091B  OFF   ON    OFF                      0     0    0
     d06d8lm35        00       OFF   ON    OFF                      0     0    0
     d06d8lm35        01 091E  OFF   ON    OFF                      0     0    0
     d06d8lm35        02 091F  OFF   ON    OFF                      0     0    0
******************************* Bottom of data ********************************
```

Figure 101. PDM MVS File Transfer Client Transfer Path Monitor panel.

This panel displays the status of every transfer task, communications task, and remote access server defined to the MVS PDM subsystem in the HDETABLE data set. Table 10 on page 120 contains a description of each field in the panel.

Table 10. PDM Event Manager Monitor Panel Fields

Field	Description
CMD	Command input area. Allowable values are: **CAN** Cancel the transaction in progess. **OFF** Change the desired state of this Transfer Path to Offline. Devices are no longer allocated. **ON** Change the desired state of this Transfer Path to Online. Devices are allocated. **CTA** Go to CTASK detail panel
Transfer Path Name	Name of this transfer path (the remote node name).
ID	Identifier for this transfer task or the path number assigned by the PDM subsystem.
Dev Nmbr	Device-number for this transfer task. This is filled only if the line represents a transfer task (it is blank for a communications task).
Curr State	Current state of this path: ON (online), OFF (offline), ACTIV (active), or UNALC (unallocate).
Desrd State	The desired state of this path (ON or OFF).
Ctask State	State of the communications task that this transfer path uses (ON or OFF).
Active Request	If Curr State is ACTIV, this is the request number of the transaction.
Xfer Type	Type of transfer request: UP=uploading, DOWN=downloading, and NONE=a non-data-transfer operation (such as a connect request).
Xfer Mbyte	Megabytes transferred.
Rate KB/s	Throughput in kilobytes per second. Sampled at 8 s intervals.
% Wait	If Curr State is ACTIV, this field contains the percent of time that the task performing the operation is in a wait state.

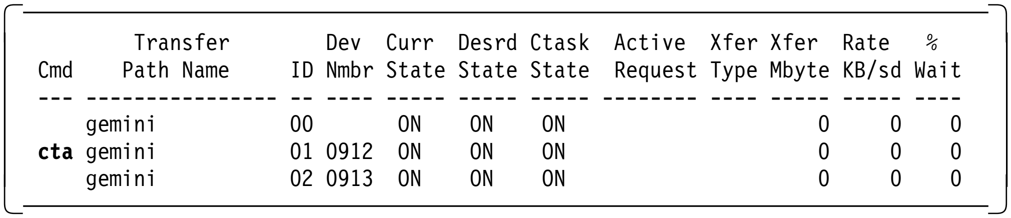
The Event Manager Monitor panel is updated every time the Enter key is pressed or a command is entered.

Your ID must be defined as a privileged ID in the HDETABLE to enter any commands on this panel. Refer to the *PRIV* statement in 6.3.1, “The DMETABLE Program” on page 51.

Notes:

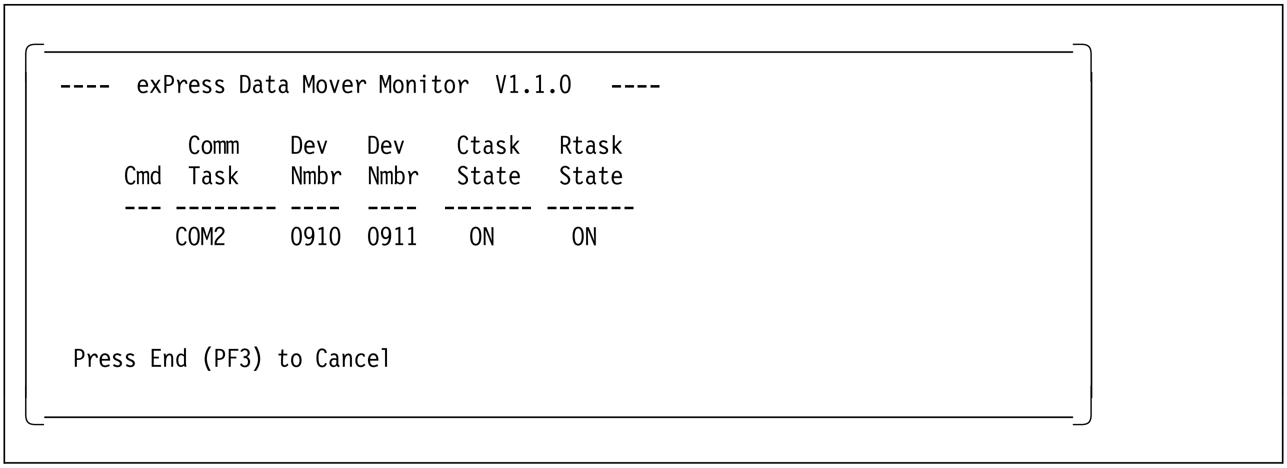
1. Cancel commands may not take effect immediately. If Curr State is ACTIV, this field contains the percent of time
2. OFF commands take effect immediately unless the task is ACTIV. When the offline function completes, the current state of the path changes to UNALC indicating the device has been unallocated by the subsystem and the desired state of the path changes to OFF.
3. Multiple commands can be entered before you press Enter.

Starting and Stopping Communication Tasks on the Event Monitor Ctask Detail Panel. Communications tasks can be started or stopped using the Event Monitor Ctask Detail panel. Entering the CTA command on any of the transfer paths listed displays the details for the communications task that is serving that transfer task. For example, if CTA is entered on any of the *gemini* transfer paths below, the Ctask detail panel for the *gemini* communications task will be displayed:



```
        Transfer        Dev  Curr  Desrd Ctask  Active  Xfer Xfer  Rate   %
Cmd    Path Name     ID Nmbr State State State  Request Type Mbyte KB/sd Wait
--- ---------------- -- ---- ----- ----- ----- -------- ---- ----- ----- ----
    gemini           00       ON    ON    ON                      0     0    0
cta gemini           01 0912  ON    ON    ON                      0     0    0
    gemini           02 0913  ON    ON    ON                      0     0    0
```

Figure 102 shows the resulting panel:



```
----  exPress Data Mover Monitor  V1.1.0   ----

          Comm     Dev   Dev     Ctask   Rtask
     Cmd  Task     Nmbr  Nmbr    State   State
     --- -------- ----  ----   ------- -------
         COM2      0910  0911     ON      ON



  Press End (PF3) to Cancel
```

Figure 102. PDM MVS File Transfer Client CTASK Detail Panel. This panel is displayed when CTA is entered on the Cmd line of a Transfer Path on the Event Manager Monitor panel.

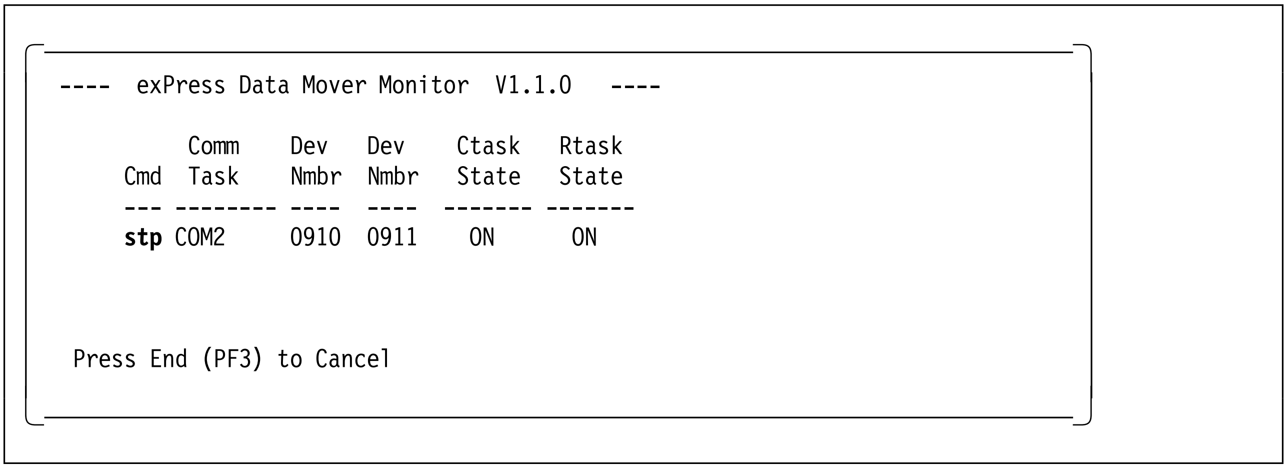
This panel displays the status of the requested communications task (Ctask) and remote access server (Rtask). Only the following commands can be entered on this panel:

STP Stop this communications task (terminate the task and free the devices). Any transfer paths associated with this communication path become unavailable.

STA Start a stopped communications task (allocate the devices).

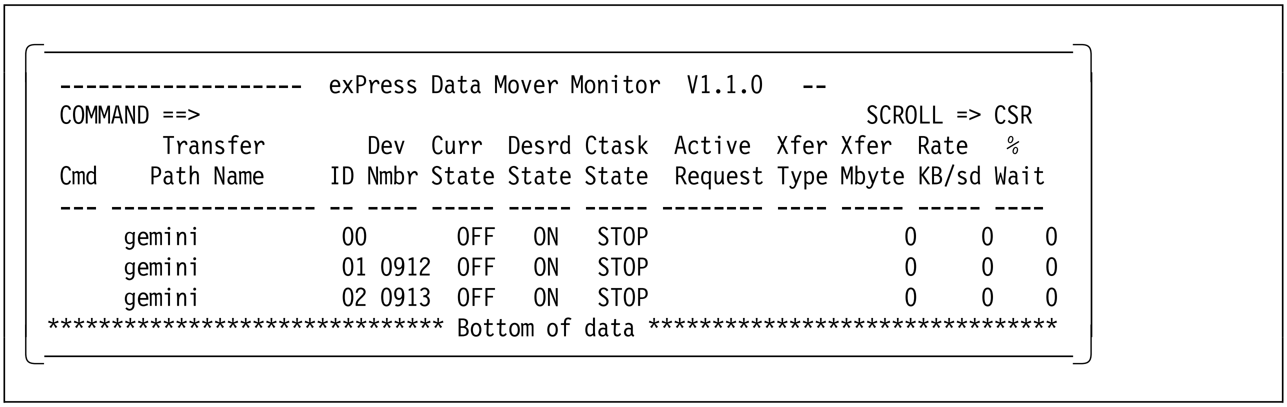
When you enter the command and press Enter, the command is executed and control returns to the prior Event Manager Monitor panel if there are no errors. Errors are displayed on this screen and remain until they are corrected or PF3 is pressed.

The Ctask panel is used to manage communications tasks only. Transfer tasks are managed from the transfer path monitor panel. Figure 103 on page 122 and Figure 104 on page 122 show the results of stopping the communications task for *gemini*



```
 ----  exPress Data Mover Monitor  V1.1.0   ----

           Comm     Dev   Dev    Ctask   Rtask
      Cmd  Task     Nmbr  Nmbr   State   State
      --- -------- ----  ----  ------- -------
      stp COM2     0910  0911    ON      ON



  Press End (PF3) to Cancel
```

Figure 103. Stopping a Communications Task from the PDM MVS File Transfer Client CTASK Detail Panel.



```
 -------------------  exPress Data Mover Monitor  V1.1.0   --
 COMMAND ==>                                                SCROLL => CSR
         Transfer        Dev  Curr  Desrd Ctask  Active  Xfer Xfer  Rate   %
 Cmd    Path Name     ID Nmbr State State State  Request Type Mbyte KB/sd Wait
 --- ---------------- -- ---- ----- ----- ----- -------- ---- ----- ----- ----
      gemini          00       OFF   ON   STOP                    0     0    0
      gemini          01 0912  OFF   ON   STOP                    0     0    0
      gemini          02 0913  OFF   ON   STOP                    0     0    0
******************************* Bottom of data ********************************
```

Figure 104. PDM MVS File Transfer Client Monitor Panel. This is the result of entering STP on the CTASK panel for remote host gemini (COM2) on the Event Monitor Panel.

As a result of stopping the communication task for gemini, the following message appears on the MVS and PDM logs:

```
 DMESYS   DME035I COMMUNICATIONS TASK COM2     HAS ENDED
 DMESYS   DME029W TRANSFER PATH gemini           00 IS UNAVAILABLE
 DMESYS   DME029W TRANSFER PATH gemini           01 IS UNAVAILABLE
 DMESYS   DME029W TRANSFER PATH gemini           02 IS UNAVAILABLE
```

To restart the communication task:

1. Enter cta on the command line of the desired communications task (any gemini ID number in this case) on the PDM Event Manager Monitor panel to display the Ctask panel.

2. Enter sta on the command line to start the communication task.

The communication task and transfer paths will initialize and establish contact with the remote node.

10.1.3 Using the PDM REXX Interface

Most users can fulfill all of their application requirements with PDM without writing programs. By using GETX or PUTX, you can tie applications to the process of data movement. The REXX-interface is provided for those users who need something more. Use the REXX interface for either of the following:

- To build MVS-driven applications that need programmatic control over the process. UNIX- and NT-driven applications can use scripts and GETX and PUTX. The REXX interface gives you the same capability on MVS.
- If you are building an integrated application and want to place PDM under the covers. This assumes that you have your own client interfaces and wish to use the high-speed PDM transport.

For REXX interface installation instructions, see the *PDM Installation and Customization Guide*, GA32-0389.

10.1.3.1 Sending and Querying Transactions

The REXX interface has two different types of calls:

- **Subsystem-calls** are commands that ask the subsystem to do work. All commands that deal with remote systems fall in this category, as do queries. The subsystem is called and a request ID may be returned. Operations can be synchronous or asynchronous.
- **Local MVS calls** are commands that involve the local MVS host only, such as LRENAME (to rename an MVS data set). The subsystem is not called to complete these operations, and they are always performed synchronously.

When you send transactions for the subsystem to process and query results, you are using only the subsystem calls. When you use the subsystem calls or the local MVS calls, you are merely adding programmatic control over file transfers.

REXX interface samples are provided in SDMESAMP.

10.1.3.2 Understanding the REXX Interface

The REXX interface runs as a function pack, which means all of the PDM functions are available to all users and can be called from any REXX script. The REXX interface then calls PDM to complete the operation.

To use the REXX interface properly:

1. Set all of the required PDM variables, which all start with ebl_.
2. Call the function, passing the proper arguments.
3. Test returned arguments to determine and display errors.

The interface is easy to use and provides a simple mechanism for scripting control over file transfers. The following is an example of a REXX script that moves a file from an NT system to MVS:

```
ebl_remote_host_name = 'ntsys' /* provide name of remote host */
ebl_remote_host_userid = 'mylogonid' /* provide my logon id on the host */
ebl_remote_host_password = 'mypass' /* provide my password on this host */
ebl_type = 'S' /* run synchronous */
ebl_mode = 'S' /* run serial */
ebl_data_type = 'A' /* do data conversion ASCII-EBCDIC */
fn = 'SYSX.TEST.DATASET' /* MVS file specification */
rfn = '/home/mylogonid/test.txt' /* NT file specification */
```

```
x = get(rfn,fn) /* go get the file */
if x = 0 then do /* did it work? */
(Success processing) /* yes */
end
else do /* no */
(Fail processing)
end
```

10.2 Using the MVS PDM Subsystem Interface

The PDM function that supports sequential file processing (BSAM, QSAM) is the subsystem interface (SSI). SSI provides the ability to execute MVS applications unmodified and access data on remote nodes. SSI supports reading and writing to PDM. This is accomplished by coding the *SUBSYS* parameter on the DD statement of the JCL or through the use of dynamic allocation.

10.2.1 Activating the SSI

To activate the SSI, the following is coded on the EXEC statement in the JCL of the started procedure:

```
//DME     EXEC PGM=DMESUPER,PARM=′SUBSYS=DMES,SSI=Y′
```

When the SSI is active, the following message is received on the operator console at subsystem startup:

```
DME064I SSI INITIALIZATION COMPLETE
```

At this point, you can run a job that uses the SSI, but a connection to the requested remote node must exist or the SSI job will receive an error message during allocation.

If the SSI is not active, upon subsystem startup the following message is received:

```
DME061W SSI SSVT CREATION FAILED WITH RC(008) AND REASON CODE(004)
```

To check the status of the SSI, issue the display SSI command.

10.2.2 Running an SSI Job

Use the SUBSYS JCL keyword or the DALSSNM dynamic allocation key to allocate the SSI to a PDM DD statement. Along with the subsystem name, other parameters such as remote host name and user ID, must be supplied with SUBSYS keyword or the DALSSPRM dynamic allocation key.

The following is an example of the JCL SUBSYS keyword:

```
//DMEDD   DD SUBSYS=(DMES,′USER=myuserid′,′PASS=mypass′,′HOST=host1′,
//        ′PATH=/home/myuserid/test.fil′)
```

In the above example, PDM uses myuserid and mypass for authorization access to the remote node or host host1. The PATH specification of /home/myuserid/test.fil is actually the file name on the remote node. The DSNAME JCL keyword can be substituted for the PATH keyword. The minimum required parameters in the JCL are:

```
USER
PASS
HOST
PATH (or DSNAME)
```

Parameters passed through JCL, profiles, or dynamic-allocation can be entered in upper or lower case except when the remote node or application is case sensitive.

10.2.2.1 Using SSI Profiles

Profiles can be created to reduce the need to specify all the keywords in your JCL. Profile information, such as userID and password, can be created for the various remote nodes you access. In your JCL, you only need to code the file name, node name, and profile name. PDM reads the profile to substitute the missing parameters.

Profiles may be members of a PDS or PDSE, or any sequential file. If you specify a profile name that is eight characters or less, PDM looks for a PDS or PDSE by the name *userid.DME.PROFILE* where *userid* is your MVS user ID. The profile name you supply is the member name in this dataset. You can also specify a fully qualified data set name, which PDM reads it as a sequential file. If this fully qualified data set name is a PDS or PDSE, the member name must be included in parentheses.

Within the profile, each keyword is coded on a separate line followed by one or more blanks, followed by the value. Comments begin with an asterisk (*) in column one or after a blank that follows a value. The following is a sample profile in PDS *userid.PROFILE.MINE* in member name *MINE*:

```
* SAMPLE PROFILE
USER     myuserid
PASS     mypass
PREFIX   /home/myuserid/
DATATYPE A
NOTIFY   mymvsid
```

In the above sample, PDM uses myuserid and mypass for authorization access to the remote node. It prefixes any file specifications with /home/myuserid/ and defaults to conversion of all files from EBCDIC to ASCII. When complete, PDM sends a message to MVS user ID myuserid. Any profile parameter can be overridden with JCL or dynamic allocation specification. The name or member name of the profile is coded in the PROFILE keyword of the SUBSYS JCL parameter:

```
//DMEDD   DD SUBSYS=(DMES,′PROFILE=MINE′,′HOST=host1′,
//          ′PATH=test.fil′)
```

Within profiles, single quotes (′) may be used to embed blanks and single quotes(′). No other embedded characters require quotes.

10.2.2.2 Using Dynamic Allocation

The SUBSYS parameter is supported in dynamic allocation by the use of two parameters, DALSSNM to specify the name of the subsystem and DALSSPRM to specify any parameter keys. For example, the following JCL statement:

```
//DMEDD   DD SUBSYS=(DMES,′PROFILE=MINE′,′HOST=HOST1′,
//          ′PATH=test.fil′,′FILETYPE=F′,′DATATYPE=B′)
```

is coded as text unit parameters for dynamic allocation (SVC 99):

```
*        ...
S99TUPL  DC    A(PDMDD)              ADDRESS OF DDNAME
         DC    A(PDMSS)              ADDRESS OF SUBSYSTEM NAME
         DC    X’80’                 INDICATE LAST TEXT UNIT
         ORG   *-1
         DC    A(PDMPRS)             ADDRESS OF SUBSYSTEM PARAMETERS
*
PDMDD    DC    AL2(DALDDNAM)
         DC    AL2(1)                 NUMBER OF PARAMETERS
         DC    AL2(L’PDMDDNM)
PDMDDNM  DC    C’DMEDD’
*
PDMSS    DC    AL2(DALSSNM)
         DC    AL2(1)                 NUMBER OF PARAMETERS
         DC    AL2(L’PDMSSNM)
PDMSSNM  DC    C’DMES’
*
PDMPRS   DC    AL2(DALSSPRM)
         DC    AL2(5)                 NUMBER OF PARAMETERS
         DC    AL2(L’PDMPROF)
PDMPROF  DC    C’PROFILE=MINE’
         DC    AL2(L’PDMHOST)
PDMHOST  DC    C’HOST=HOST1’
         DC    AL2(L’PDMPATH)
PDMPATH  DC    C’PATH=test.fil’
         DC    AL2(L’PDMFILT)
PDMFILT  DC    C’FILETYPE=F’
         DC    AL2(L’PDMDATT)
PDMDATT  DC    C’DATATYPE=B’
*        ...
```

10.2.2.3 SSI Sample - UNIX to UNIX Data Transfer

Figure 105 presents sample JCL to send a file from UNIX host *gemini* to UNIX host *kingpin* using the SSI:

```
//SENDUNIX JOB ,’PDMSYS’,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=H,CLASS=A,NOTIFY=userid
//*******************************************************************
//*        USE THE SSI W/IDCAMS TO COPY A UNIX FILE TO ANOTHER UNIX   *
//*******************************************************************
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//UNIXIN   DD  DSN=’/tmp/testfile.tst’,
//             DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//             SUBSYS=(DMES,’HOST=gemini’,’USER=user1’,
//             ’PASS=password’,’FILETYPE=F’,’DATATYPE=A’)
//UNIXOUT  DD  DSN=’/tmp/testfile2.tst’,
//             DCB=*.UNIXIN,
//             SUBSYS=(DMES,’HOST=kingpin’,’USER=user1’,
//             ’PASS=password’,’FILETYPE=F’,’DATATYPE=A’)
//SYSIN    DD  *
  REPRO  INFILE(UNIXIN) OUTFILE(UNIXOUT)
/*
```

Figure 105. Sending Data from UNIX to UNIX Using SSI

In the example in Figure 105 any utility that can copy any sequential file could have been used (for example IEBGENER).

On the input DD statement, the file to be copied is specified using the DSN/DSNAME parameter. The file name can also be specified using the PATH subparameter of the SUBSYS parameter.

On the input DD statement, you must supply DCB information such as LRECL, BLKSIZE, and RECFM if the utility or the program does not supply this information. As files in a UNIX environment are not grouped in records or blocks the values specified instead tell the utility how many bytes to handle at a time.

The SUBSYS parameter contains the PDM subsystem name and parameters to be passed to PDM. The HOST, USER, and PASS subparameters tell which system (node) the data is stored on and which userID is to copy it. The FILETYPE subparameter specifies that the input is a file (F). The DATATYPE subparameters request PDM to translate the data from ASCII to EBCDIC before it is given to IDCAMS.

On the output statement, the file into which the data should be copied is specified. The DCB information must once again be specified if the utility does not automatically copy this information from the input statement.

See Chapter 12, “Data Copy Sharing Samples” on page 141 for more JCL samples using the PDM MVS SSI.

10.3 MVS Data Sets

The file transfer clients all support SITE (node clients) or LOCSITE/LSITE (MVS client) for the specification of new data set allocation parameters. If you cause a data set to be written on MVS, PDM uses the presence of any SITE or LSITE parameter to specify that you intend to create a new data set. PDM calls dynamic allocation and passes the user′s specifications for the new data set. If PDM creates a new data set and an error occurs during the transfer, PDM automatically deletes and uncatalogs the data set unless checkpointing is active. However PDM automatically deletes only data sets that it creates and that are in error.

PDM operates in both SMS and non-SMS environments. You may also use automatic class selection (ACS) routines to fill in missing allocation parameters. Any parameter explicitly coded on a SITE or LSITE statement is passed to dynamic allocation. PDM cannot determine whether your specification is complete. If it is not, you receive a dynamic allocation error message or an OPEN error message for the data set.

Allocation errors appear as return code of 227 from PDM. The reason code contains the dynamic allocation return code. When decoding the dynamic allocation return code for you, PDM calls IEFDB476, which produces a formatted message describing the problem. Whenever you get a return code of 227 from PDM, you also receive this formatted response. For further information, look up the error code in the *OS/390 MVS Authorized Assembler Services Guide*, GC28-1763.

OPEN errors cause PDM to end abnormally when opening the data set (ABEND S013). These errors are almost always user errors. PDM recovers from the error and posts an error code of 238. Look on the system log to find the S013 error along with the IEC-message. To determine the cause of the error, look up the IEC-message in the *OS/390 MVS System Messages Volume 4*, GC28-1787. For example, if you fail to specify BLKSIZE and it is not supplied by the open function, when attempting to open (with OPEN) the new data set for output, PDM receives an S013 ABEND.

Table 11 contains a list of the allocation parameters that PDM supports and the keyword used on the SITE or LSITE statement to specify each.

Table 11. PDM SITE and LSITE Parameters

Keyword	**Meaning**	**JCL Equivalent**
BLKSIZE	Block size	BLKSIZE=
BLOCKS	Allocation is in block units	SPACE=< nnn,...
CLEAR	Clear all SITE specifications	
CYLINDERS	Allocation is in cylinder units	SPACE=< CYL,...
DATACLAS	SMS data class specification	DATACLAS=
DIRBLKS	Number of directory blocks (PDS)	SPACE=(..,(..,..,nn))
DISK	Create a data set on DASD	
EXPDT	Expiration date	EXPDT=
LRECL	Record length	LRECL=
MGMTCLAS	SMS management class specification	MGMTCLAS=
MODEL	Model DSCB specification	DCB=dsname
NUMUNIT	Number of units to allocate	UNIT=(...,n)
OFF	Turn off site specifications (retain values)	
ON	Turn on site specifications	
PRIMARY	Primary allocation value	SPACE=(...,(n,....
RECFM	Record format	RECFM=
RETPD	Retention period	RETPD=
SECONDARY	Secondary allocation value	SPACE=(...,(...,n...
STORCLAS	SMS storage class specification	STORCLAS=
TAPE<n>	Create a data set on TAPE of n volumes	VOL=(,,,n..
TRACKS	Allocation is in track units	SPACE=(TRK,.....
UNIT	UNIT specification	UNIT=
VOLUME	VOLUME specification	VOL=SER=

For output tape volumes, it is not enough to specify only the UNIT parameter. Because the subsystem needs to know that tape output is requested, a SITE TAPE statement must also be included. SITE TAPE 7 specifies a maximum of seven output volumes. If the statement is coded as merely SITE TAPE, the system default (5) is assumed. To revert to non-tape usage, code LSITE DISK. For more details, see 10.3.5, “Using Tapes” on page 131.

SITE parameters remain in effect for the duration of the invocation of the client, but they are used only if you are performing an operation that would cause the creation of a new data set. For example, if you specified SITE STORCLAS ABC, and you perform a GET using a node client, the parameter is ignored, because a GET retrieves an existing data set.

SITE parameters can be turned off temporarily for a transfer operation by specifying SITE OFF. You may later reactivate the specifications by coding SITE ON.

Note: The VERIFY function displays the current settings of all SITE parameters.

SITE CLEAR resets all parameters to Not Supplied. Any specific parameter can be reset by entering the SITE statement with the parameter and no value. For example, to turn off the STORCLAS parameter, code SITE STORCLAS.

All new MVS data sets are cataloged by PDM (even without SMS). If PDM has created the data set and an error occurs, the data set is removed from the catalog unless checkpointing is active.

Note: You can create only partitioned and sequential data sets with PDM.

You cannot create new VSAM data sets, even if your SMS parameters will allocate it. PDM does not use VSAM to write to the data set. You cannot create a new GDG base entry with PDM, but you can create new generations of an existing base. Refer to 10.3.3, "Using Generation Data Groups" on page 130 for more information.

10.3.1 PDS and PDSE Specifications

Partitioned data sets (PDS) and partitioned data sets extended (PDSE) are supported. For a PDS or PDSE, code the member name in the data set specification with parentheses. The local current working directory can specify the name of the PDS or PDSE, and the command can contain only the member name. For example, the following copies the member MEMA in (PDS or PDSE) MYLIB.TEST.PDS to /home/test/file1 on the remote node:

```
lcd /mylib.test.pds
put (mema) /home/test/file1
```

This could also be coded in one statement:

```
put mylib.test.pds(mema)  /home/test/file1
```

The following sample sequence lists the members of a PDS or PDSE (in an interactive session) and then renames a member:

```
DMEFTP==> lcd /mylib.test.pds MYLIB.TEST.PDS.
DMEFTP==> ldir (*)
    Name         VV.MM  Created       Changed     Size   Init  Mod Id
MEMA        01.06  10/24/95  10/24/95   15:13    52   52     0  BXXABCD
MEMB        01.26  01/13/95  11/29/95   14:28    63   47    34  BXXABCD
DMEFTP==> lrename (mema) (memc) DME544I MEMBER MEMA RENAMED TO MEMC
```

You can use wildcards to list members of a PDS or PDSE. Use an asterisk (*) to specify any number of characters and a percent symbol (%) to indicatea single wildcard character. For example, to list all members that begin with abc, code abc*. To list all members that are six characters long and end in xyz, code %%%xyz.

To create a new PDS and add the first member, code all SITE parameters (including DIRBLKS) and then get (with GET) the first member. You cannot create a PDSE using PDM.

The only difference between PDS and PDSE as handled by PDM is in the DISP of the data set. MVS supports concurrent writing to PDSE, but not to PDS. Therefore, if you submit a PARALLEL transfer to multiple members of a PDS, PDM serializes the requests (runs them one at a time) using DISP=OLD. Multiple transfers in PARALLEL to a PDSE are not serialized and PDM uses DISP=SHR.

10.3.2 Using Model Data Set Control Blocks

You can use model data set control blocks (DSCBs) when creating any new MVS data set with PDM. You must specify the full data set name on the SITE MODEL statement. Even if you have issued a change directory (cd) command to a different MVS data set level, you must still code the entire data set name on the statement. Model DSCBs are required if you create new generation datasets in a non-SMS environment.

10.3.3 Using Generation Data Groups

For generation data groups (GDGs), code 0 (zero) after the data set name to refer to the most recent level, -n to refer to prior levels, and +1 when creating a new level. New levels require the full complement of SITE parameters.

Note: For GDGs, the current level (0) changes immediately after the dataset has been created.

PDM does not automatically create the GDG base entry; however, you can create new generation levels according to the SITE specification. Non-SMS systems require a model DSCB, specified with the SITE MODEL statement.

For example, to copy the most current version of a GDG named MYLIB.TEST.GDG to remote file /home/user/file1, code:

```
put mylib.test.gdg(0)  /home/user/file1
```

The PDM subsystem serializes GDG requests using the base GDG name, so that multiple transfers to the same name all run successively and generate new GDG levels. For example, the following commands both generate new levels (assuming SITE parameters have been defined for the new data sets):

```
get /home/user/file1 mylib.test.gdg(+1)
get /home/user/file2 mylib.test.gdg(+1)
```

If you then decide to retrieve /home/user/file1, code the following:

```
put mylib.test.gdg(-1)  /home/user/file1
```

The current level 0 contains /home/user/file2.

Be aware that you cannot run a multistep job that calls the PDM file transfer client to create a new GDG member, followed by a step that accesses that member. MVS allocation resolves GDG names at job initiation and will not know that PDM has created a new generation. The following is an alternative:

1. Call PDM synchronously.

2. Use an IEBGENER step to copy a job to the internal reader, which is the step you intended to run second.

When the second step is scheduled, PDM has completed and the generation levels are correct.

10.3.4 Virtual Storage Access Method (VSAM) Data Sets

PDM can read any Virtual Storage Access Method (VSAM) data set, but PDM can not write VSAM data sets. Refer to a VSAM data set in the same manner as any sequential data set when creating specifications for the PDM clients. PDM uses the catalog to determine that it is a VSAM file and uses the VSAM access method. You can transfer in BINARY, ASCII, or ASCII NOTRIM mode. Empty records in RRDS format files are not transferred.

Checkpoint/restart is not supported for VSAM data sets.

10.3.5 Using Tapes

PDM can create or read any sequential file on tapes, physical or virtual, that have IBM standard labels. For reading, PDM uses the system catalog to find the data set. Volume mounting is handled by the MVS system (and tape management system). For new data sets, PDM must know that it is creating a tape data set. You must code SITE TAPE in addition to any SITE UNIT statement to tell PDM that this is a tape data set.

PDM defers mounting of tape volumes until OPEN, but it checks to see if units are available before running the transaction. In this manner, your transaction may be queued by the subsystem pending availability of drives.

Some tape management systems use the EXPDT value to determine tape disposition. PDM passes any specification for SITE EXPDT unmodified to the tape management systems; however, be aware that PDM uses the long form interface to dynamic allocation (DALEXPDL).

10.3.6 Migrated Data Sets

PDM allows migrated data sets to be recalled before moving files to or from them. PDM does not recall PDS and PDSE for listing purposes. PDM uses the presence of MIGRAT in the volume serial number to determine that a data set is migrated. A transaction that refers to a migrated data set is not queued.

10.3.7 Directory Emulation

MVS data set names consist of qualifiers that are somewhat similar to directories. Qualifiers may be no longer than eight characters and are separated by a period (.). The total number of characters may not exceed 44. Each qualifier must start with a letter (A-Z), at sign (@), number sign (#), or dollar sign ($). The remaining characters in a qualifier name may also include numbers. All MVS data set names are automatically converted to upper case.

PDM clients simulate directories for these qualifiers. For example, if you are using the node file transfer client and you want to refer to all data sets that begin with SYS1, code the following:

```
cd /sys1
```

Subsequently, all data sets that begin with SYS1.LIB (and so forth):

```
cd lib
```

The file transfer clients prefix the local current working directory to all functions that refer to MVS data sets unless the data set name begins with a slash (/). For example, if MYLIB.DATA is the current working directory and you enter the following command to the node file transfer client, the data set MYLIB.DATA.XYZ will be deleted:

```
delete xyz
```

When changing the current working directory, you may specify a complete new prefix starting with slash (*cd /new*), a relative path (*cd next*), or change up one directory (*cd..*).

When using the full data set listing feature (assuming you have the Catalog Search Interface installed), the current working directory determines what appears on the listing. The current working directory is removed from the list of data sets. The following command sequence shows the difference between listing the same data sets with and without the use of a current working directory:

```
DMEFTP==> cd /sys2.mylib
DMEFTP==> dir
Requesting dataset list for ′SYS2.MYLIB.*′
Request ID is ′17500312′
Volume Unit          Date Ext    Used Recfm Lrecl BlkSz Dsorg Dsname
MIGRAT                                                    CKPT
STGD14 3380       04/24/97 01     01   VB     150   154  PS   DATA
STGD02 3380       01/24/97 01     01   VB   32756 32760  PS   TABLE
STGD02 3380       04/15/97 01     01    F      80    80  PS   CODE
Request ′17500312′ completed successfully
Directory complete
DMEFTP==> cd /
′.′  is the remote (MVS) Directory
DMEFTP==> dir /SYS2.MYLIB.*
Requesting dataset list for ′SYS2.MYLIB.*′
Request ID is ′17500313′
Volume Unit          Date Ext    Used Recfm Lrecl BlkSz Dsorg Dsname
MIGRAT                                                  SYS2.MYLIB.CKPT
STGD14 3380      04/24/97 01     01   VB      150   154 PS  SYS2.MYLIB.DATA
STGD02 3380      01/24/97 01     01   VB    32756 32760 PS  SYS2.MYLIB.TABLE
STGD02 3380      04/15/97 01     01    F       80    80 PS  SYS2.MYLIB.CODE
Request ′17500313′ completed successfully
Directory complete
DMEFTP==>
```

10.4 Piping Data Using GETX/PUTX Commands

You can establish a direct logical pipeline from the downstream platform to the OS/390 platform with data flowing transparently through the intermediate platform. PDM client nodes provide the same data piping and query functions provided on channel-attached platforms. In the example shown in Figure 106 on page 133 the output from the UNIX-utility *tar* is send to a tape-file on OS/390 instead of being written to a local tape drive.

```
//DMEGETX JOB ,’PDMSYS’,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=H,CLASS=A,NOTIFY=HSMATH1
//STEP01  EXEC PGM=DMEFTP,PARM=(’SOURCE(AUTOLOG)’)
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//DMECLIB  DD  DSN=PDM.SDMESRCE.CNTL,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  *
   lsite tape
   lsite unit vtstape
   lsite recfm vb
   lsite lrecl 32756
   lsite blksize 32760
   getx /etc/sbin/tar backup.tape.dataset -xrv /dir/file/source.fil
   close
/*
```

Figure 106. Piping Data Using GETX Command

10.4.1 How to Pipe Data on MVS

IBM exPress Data Mover offers high-speed data piping using the subsystem interface and requires no programming changes. With SSI, simple Job Control Language changes redirect the data flow from the intermediate disk or tape storage to the PDM server software on the S/390. PDM then flows the data directly to the UNIX or NT server PDM, which then passes the data to the specified executable using standard pipe functions.

Piping of data between job steps in OS/390 is available through BatchPipes or SmartBatch. For more information on BatchPipes and SmartBatch, see *BatchPipes/MVS V1R2 Introduction*, GC28-1214, or *SmartBatch V1R2 Overview*, GC28-1627. Neither BatchPipes nor SmartBatch is required in order to pipe data to PDM.

For example, directly coupling database extraction programs on a sending server with database-load programs on a receiving server, as shown in Figure 107, eliminates the extra time and cost required to create separate intermediate flat files on each server before loading the extracted data into the other database.

```
//GETMONEY JOB (acct)
//GETIT   EXEC PGM=EXTRACT
//SYSPRINT DD SYSOUT=*
//SYSUT1  DD DSN=DBDEPT.EXTRACTS.CURRENCY.HOURLY,DISP=SHR
//SYSUT2  DD SUBSYS=(BP01),DSN=SAMPLE.BATCH.PIPE.MVS,
//      DCB=(RECFM=..??..,LRECL=..??..)
//* SYSUT2 USING BATCH PIPING
//FIXUP   EXEC PGM=CLEANSE
//INPUT   DD SUBSYS=(BP01),DSN=*.GETIT.SYSUT2,
//      DCB=*.GETIT.SYSUT2
//OUTPUT1 DD SUBSYS=(DMES,’PROFILE=BANK,HOST=CURRENCY2’,
//           ’PATH=/usr/trading/dbload’,
//           ’FILETYPE=X’,’FILEARGS=..??..’)
```

Figure 107. Piping Data To Database Loader

Chapter 11. Performance Expectations

Using InfoSpeed, you can implement cost-effective, high-performance solutions for applications such as e-business and business intelligence.

InfoSpeed is a storage-based data-copy sharing solution that copies data between servers using ESCON channels and, if necessary, SCSI bus connections rather than networks. Data is replicated from one server to another without the use of intermediate flat files.

Dedicated ESCON/SCSI or ESCON-only connections to transfer data are more cost effective than networks at up to 10 times the speed of networks. Bulk data transfers are offloaded from existing networks, allowing them to more optimally service transactions without becoming overloaded. Server processing cycles are significantly lower for storage-based data transfers than for network-based transfers.

InfoSpeed supports multiple parallel transfers of binary and coded data in either direction between connected servers. PDM converts code between S/390 data format and UNIX or NT format.

11.1 Performance Expectations Using the Data Gateway

Performance for InfoSpeed using the Data Gateway hardware in conjunction with PDM software is summarized in Table 12.

Table 12. InfoSpeed Performance

Environment	**ESCON to SCSI Data Transfer Rate**
Multiple files, single server	Up to 14 MB/s
Multiple files, multiple servers (2 ESCON x 2 SCSI; no IBM 9398-MX1)	Up to 28 MB/s Up to 14 MB/s/path
Multiple files, multiple servers (2 ESCON x 12 SCSI; with IBM 9398-MX1)	Up to 26 MB/s Up to 13 MB/s/path

File transfers are as fast as the slowest element involved in the transfer. Typically, this is a function of the storage device on one side or the other. The rates quoted in Table 12 reflect memory-to-memory transfers to avoid storage device interference. The highest rates are achieved by running multiple transfers in parallel so that multiple connections are involved on both the SCSI and the ESCON sides.

Use of an IBM 9398-MX1 to enhance connectivity adds overhead to the transfer path. The IBM 9398-MX1 was designed to impose less than 8% performance degradation in order to minimize its performance impact.

11.2 Performance Expectations for ESCON Only

Performance for PDM in an ESCON-only environment is summarized in Table 13.

Table 13. PDM ESCON Only Performance

Environment	ESCON only Data Transfer Rate
Multiple files, single server	Up to 13 MB/s

Again, the rate quoted in Table 13 reflects memory-to-memory transfers to avoid storage device interference. Performance when the IBM 9398 Data Gateway hardware is not required and the UNIX (today AIX only) server has ESCON channel attachment capability is shown in the right-hand column. This observation is based on the use of an RS/6000 with an ESCON adapter card installed and shows the results for multiple files in parallel using a single ESCON channel connection between the S/390 server and the RS/6000 server.

Disclaimer: Measured performance data contained in this document was determined in a controlled environment. The results obtained in other operating environments may differ significantly.

11.3 Performance Considerations

InfoSpeed transfers data as quickly as possible. Transfer speed for a single file of data is limited by the slowest element involved in the transfer. Typically, that is the slowest of the storage devices used on the sending and receiving ends. Higher overall transfer rates are achieved by running multiple transfers in parallel. The number of transfer paths also directly affects the transfer rate. Use of two ESCON and two SCSI ports on IBM InfoSpeed Data Gateway with two ESCON channels on S/390 servers and two SCSI ports on UNIX or NT servers provides twice the data transfer capacity of a single transfer path.

11.4 Performance in Terms of Data Transfer Rates

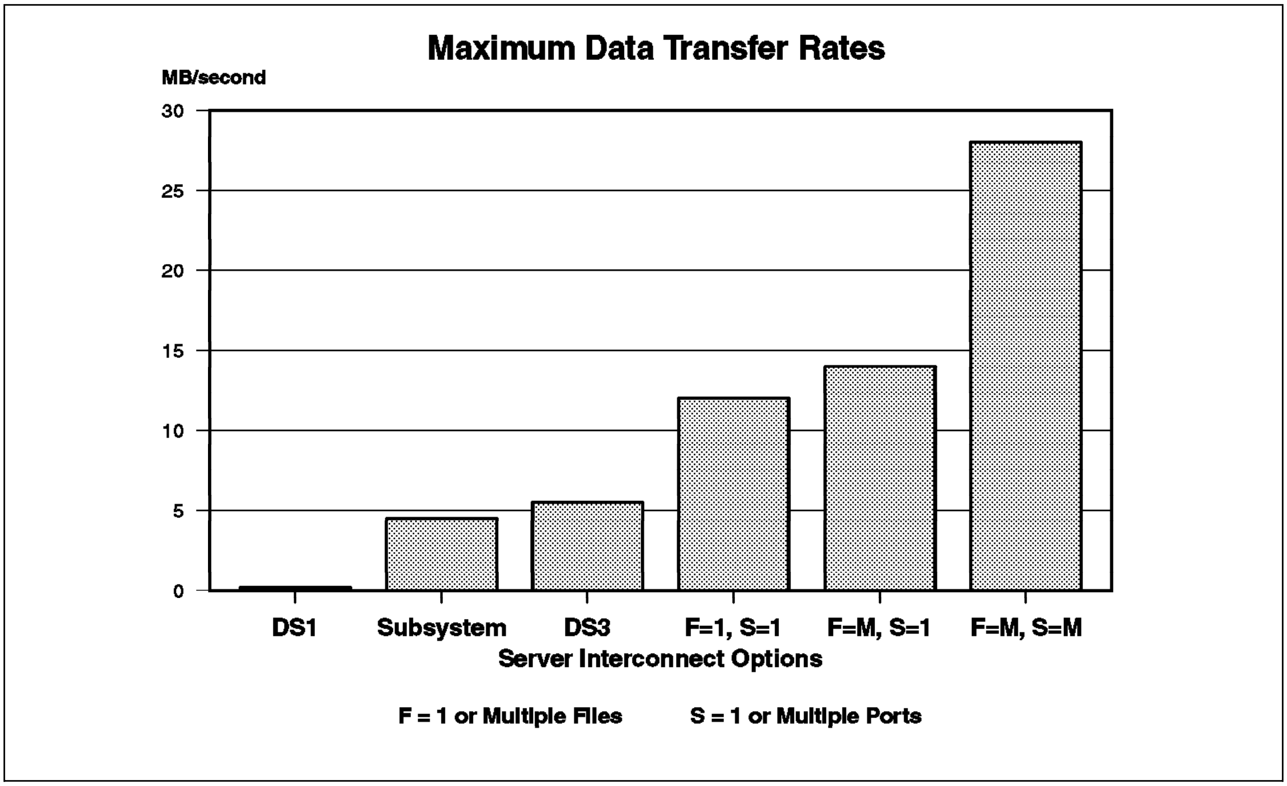
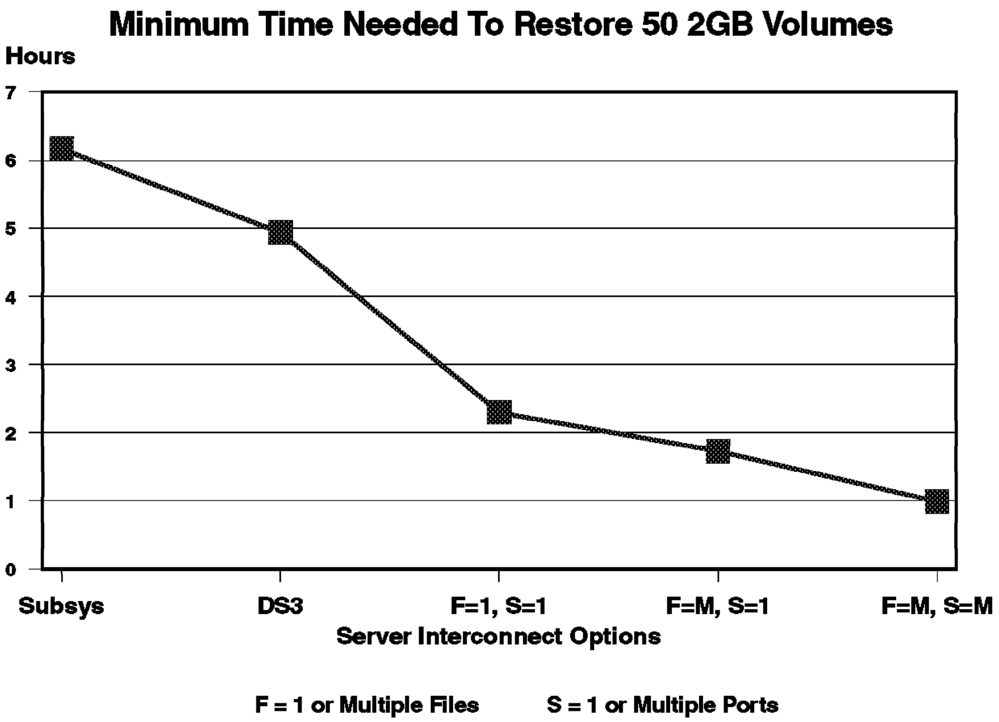


Figure 108. Maximum Data Transfer Rates for Selected Server Interconnect Options

Figure 108 is intended to show the relative throughput rates of various server interconnect options. It is somewhat unrealistic in that both telecommunications options are depicted without any protocol delays. The data transfer rate of the DS1 or T-1 link is 193 KB/s. The data transfer rated of the subsystem-based method is reported to be between 4 and 5 MB/s (4.5 MB/s is used in the example). The data transfer rate of the DS3 or T-3 link is 5.5 MB/s. The data transfer rate of 12 MB/s represents one file transfer between one S/390 server over one ESCON channel through an IBM InfoSpeed Data Gateway connected to one open-system server through one SCSI bus. The data transfer rate of multiple parallel file transfers between one S/390 server over one ESCON channel through an IBM InfoSpeed Data Gateway connected to one open-system server through one SCSI bus is 14 MB/s. The data transfer rate of multiple parallel file transfers between one or two S/390 servers connected over two ESCON channels through an IBM InfoSpeed Data Gateway connected to one or two UNIX/NT systems servers over two SCSI buses is 28 MB/s.

11.5 Performance in Terms of Elapsed Time



Server Interconnect Options

Figure 109. Minimum Time Needed to Restore 50 2-GB Volumes for Selected

Figure 109 is intended to show the relative speed required to restore fifty 2-GB volumes (100 GB). The amount of time required to restore the volumes over a T-1 line is not shown, since it is greater than one day and inclusion of that point on the chart severely distorts the relative difference between the other options. The essence of this chart is that IBM′s InfoSpeed solution connected using two ESCON channels and two SCSI buses can restore the fifty volumes far faster than any similar option.

11.6 Increasing Bandwidth or Parallelism for a Server

In certain configurations the six data paths available on a single SCSI interface might not be enough. The bandwidth one SCSI interface offers, 13-14 MB/s, for some installation might not be sufficient. IBM exPress Data Mover supports multiple interfaces to a single server or node. As each node needs only two control paths, and they already have been defined on the first SCSI interface, all eight LUNs on the second interface can be used as data paths. In Figure 110, a configuration offering a total bandwidth of 56 MB/s is shown.

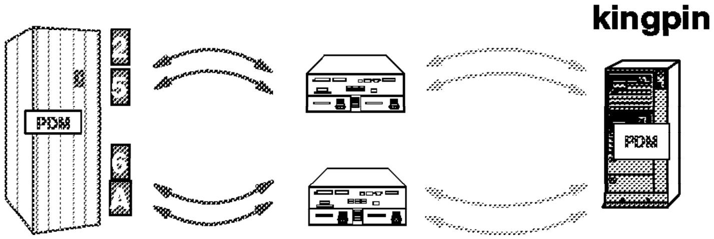


Figure 110. Example of an Increased Bandwidth Configuration

When a configuration like that shown in Figure 110 is installed you can distribute the workload across the available paths by specifying the device numbers in the order you want PDM to allocate the data paths (See Figure 111).

```
kingpin   hdef iodrv=dmeiocts,comm=kingcom,unit=(202,2),unit=(602,2),
               unit=(502,2),unit=(A02,2),unit=(204,2),unit=(604,2),
               unit=(A04,2)
kincom    cdef unit=200,ctdrv=dmectcts,type=ctc,ras=y
```

Figure 111. Device Number Distribution

Chapter 12. Data Copy Sharing Samples

This chapter describes simple examples how to start using InfoSpeed with basic configurations.

12.1 Connecting Options

There are several connection options to use when copying data between supported UNIX/NT (open) and S/390 environments. Also, copying from one open environment to another open environment via S/390 is a possibility. Figure 112, describes the connecting possibilities when the IBM InfoSpeed solution was made generally-available in June 1998.

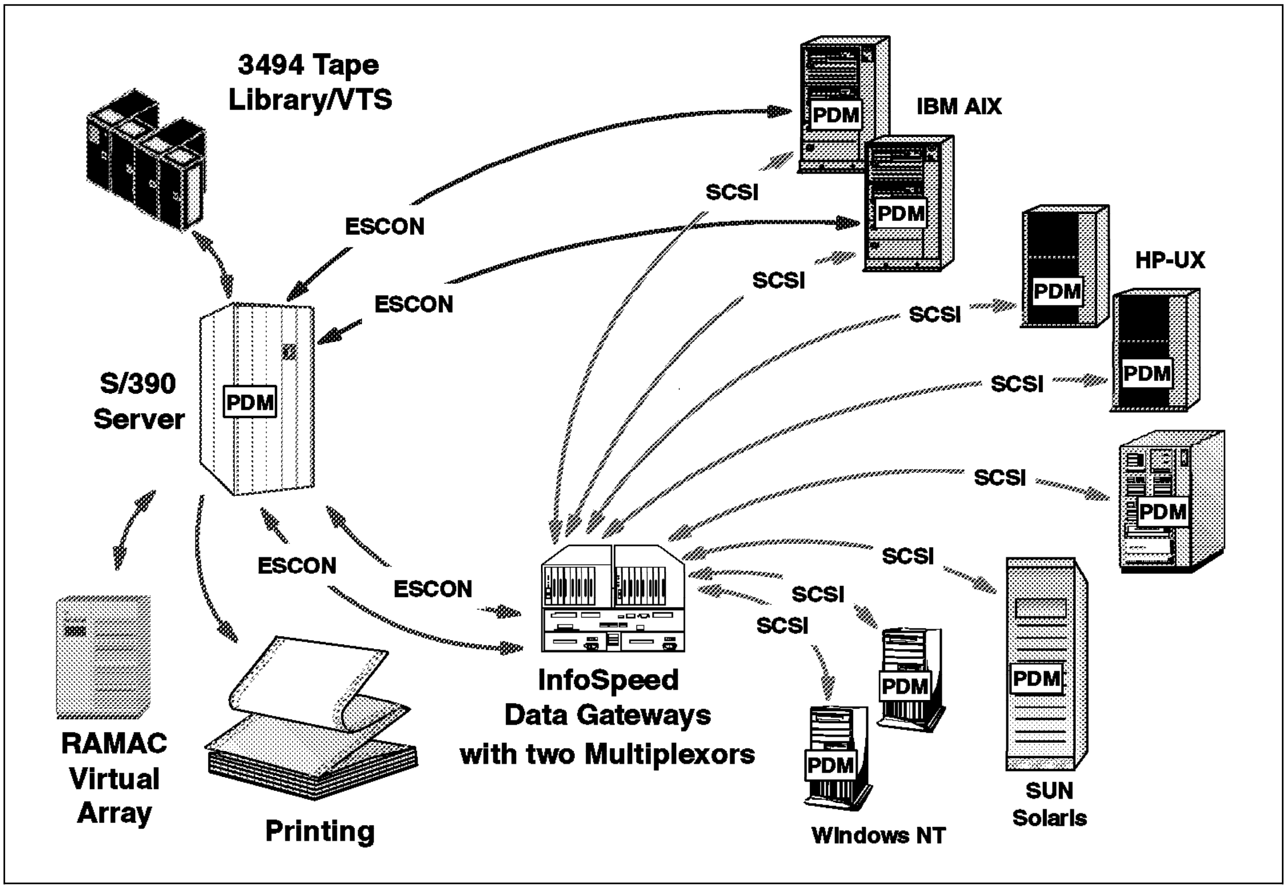


Figure 112. Connecting Options for Data-Copy Sharing

12.2 Data-Copy Sharing Samples Using MVS

These samples show how to initiate the data copy movement using the MVS interface. There are four ways to do it:

- The file transfer client
- The ISPF client
- The REXX function calls
- The JCL and SSI

12.2.1 Using the File Transfer Client

12.2.1.1 Copying a UNIX file to MVS (GET)

Figure 113 is an example of JCL to create a new data set on MVS using the *dataclas* lsite parameter. In this example, a data class called *PSEQ0001* was previously defined in MVS under DFSMS. Data class PSEQ0001 defines the data set′s characteristics, which match the lsite parameters from the example in Figure 88 on page 107.

```
//BUJOBB   JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  *
* Open remote node named ″gemini″
   open gemini
* Turn off echoing so password does not appear in output
   echo off
* Identify userID and password to ″gemini″
   log user01 mypasswd
* Turn echoing back on to display commands
   echo on
* Specify mvs new data set dataclass parameter
   lsite dataclas pseq0001
   lsite unit sysda
   notify mymvsid
* Set remote current working directory
   cd /home/user01
   ASCII
* Copy file ″/home/user01/myfile.txt″ to new MVS data set ″myfile.user.text″
   get myfile.txt myfile.user.text
* Close the connection to ″gemini″
   close
* End the job
   quit
/*
```

Figure 113. Sample JCL for PDM MVS File Transfer Client Using GET Command with Dataclass

12.2.1.2 Copying a MVS data set to UNIX (PUT)

Data can also be copied to a remote node from MVS using JCL. Figure 114 shows the JCL and DMEFTP control statements to copy a sequential MVS data set, *mydata.test.file*, to a new file *test.fil*, in a new subdirectory */home/user01/tmp*, on a remote node *gemini*.

```
//MYJOB    JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP,PARM=PARALLEL
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  *
* Open remote node named ″gemini″
   open gemini
* Turn off echoing so password does not appear in output
   echo off
* Identify userID and password to ″gemini″
   log user01 mypasswd
* Turn echoing back on to display commands
   echo on
* Go to directory ″home″
   cd /home
* Go to subdirectory ″user01″
   cd /home
* Make a new directory ″/home/user01/tmp″
   mkdir tmp
* Go to the new directory
   cd tmp
* Set data conversion to ascii
   ascii
* Write a message on the Master Console when copy is complete
   notify operator
* Copy MVS data set ″mydata.test.file″ to ″/home/user01/tmp/test.fil″
   put mydata.test.file test.fil
* Close the connection to ″gemini″
   close
* End the job
   quit
/*
```

Figure 114. Sample JCL for PDM MVS File Transfer Client PUT command

12.2.1.3 Copying a UNIX File To a MVS Tape Data Set

The example in Figure 115 copies a UNIX binary file to a physical or virtual 3490 cartridge on MVS. The data set is to be cataloged to the volume serial number of the cartridge mounted upon successful completion of the PDM request.

```
//MYJOB JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  *
   open gemini
   log pdmsys thuy01
   lsite tape
   lsite unit 3490
   lsite recfm vb
   lsite lrecl 32756
   lsite blksize 32760
   notify hsmath1
   binary
   cd /usr/tmp
   get myfile.bin myfile.tape.bin
   close
   end
//
```

Figure 115. Sample JCL for PDM MVS File Transfer Client Copy from UNIX to a MVS Tape

12.2.1.4 Copying Multiple UNIX Files to MVS in Parallel

The example in Figure 116 is a sample job to copy multiple files in parallel within one script.

```
//MYJOB JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  *
* Open remote node named ″gemini″
   open gemini
* Turn off echoing so password does not appear in output
   echo off
* Identify userID and password to ″gemini″
   log user01 mypasswd
* Turn echoing back on to display commands
   echo on
* Move to ″/usr/tmp″ directory on ″gemini″
   cd /usr/tmp
* Set my default MVS directory
   lcd /myuser
* Copy 3 files to new MVS data sets
   lsite lrecl 80
   lsite blksize 800
   lsite recfm fb
   lsite tracks
   lsite primary 10
   lsite secondary 5
   lsite unit sysda
   notify mymvsid
   ASCII
   parallel
   get myfile1.txt test.txt001
* Change secondary allocation for ″myuser.test.txt002″
   lsite secondary 10
   get myfile2.txt test.txt002
* Turn off lsite parms because data set ″myuser.test.txt003″ exists
   lsite off
   get myfile3.txt test.txt003
* Close the connection to ″gemini″ and end job
   close
   end
//
```

Figure 116. Sample JCL for PDM MVS File Transfer Client Copying Multiple Files from UNIX

12.2.1.5 Copying Multiple Members from a PDS to a Remote Node

The following example copies many members from a PDS to the remote node *gemini* in parallel. The directory for the new files is created synchronously (PARM=SYNC in EXEC statement) because it must exist before any files are copied. The script then changes to parallel mode to actually copy the files. If the MKDIR fails, the job terminates at that point. The QUERY (Q) RESULT ALL command is used to synchronize the results and set the return code. It synchronizes all requests submitted and waits for them to complete before returning status.

```
//BAJOBA   JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP,PARM=SYNC
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  *
* Open remote node named ″gemini″
   open gemini
* Turn off echoing so password does not appear in output
   echo off
* Identify userID and password to ″gemini″
   log user01 mypasswd
* Turn echoing back on to display commands
   echo on
* Move to ″/tmp″ directory on ″gemini″
   cd /tmp
* Make directory ″pdsfiles″ on ″gemini″
   mkdir pdsfiles
* Switch to ″parallel″ mode
   parallel
   cd pdsfiles
   ascii
   notify mymvsid
* Set my default MVS directory
   lcd /mylib.pds
   put (mem1) mem1.txt
   put (mem2) mem1.txt
   put (mem3) mem1.txt
   put (mem4) mem1.txt
   put (mem5) mem1.txt
* Synchronize at this point and set the return code
   query result all
   close
   end
//
```

Figure 117. Sample JCL For PDM MVS File Transfer Client Copying Multiple Members to UNIX

12.2.1.6 Writing a File on S/390 from a UNIX Application

In the example, shown in Figure 118, a sample UNIX application (tar) writes its output to an MVS data set instead of creating a local file on the UNIX system. The sample application, the UNIX utility *tar*, archives data in compressed format.

```
//TARJOB   JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  *
* Open remote node named ″gemini″
  open gemini
* Turn off echoing so password does not appear in output
  echo off
* Identify userid and password to ″gemini″
  log user01 mypasswd
* Turn echoing back on to display commands
  echo on
  notify mymvsid
* Set allocation parameters
  lsite tape
  lsite unit 3490
  lsite recfm vb
  lsite lrecl 32756
  lsite blksize 32760
  lcd /mymvsid
  cd /etc/sbin
  getx /etc/sbin/tar source.fil.archive(+1)                            x
       -xrv /usr/user01/files/source.fil
  close
  end
//
```

Figure 118. Sample JCL for PDM MVS File Transfer Client to Write an MVS Data Set from a UNIX Application

12.2.1.7 Reading a File on S/390 from a UNIX Application

In the example, shown in Figure 119, a sample UNIX application (dd) reads its input from an MVS data set instead of a local file on the UNIX system. The sample application is the UNIX utility *dd*, which is a copy utility like IEBGENER in the MVS world.

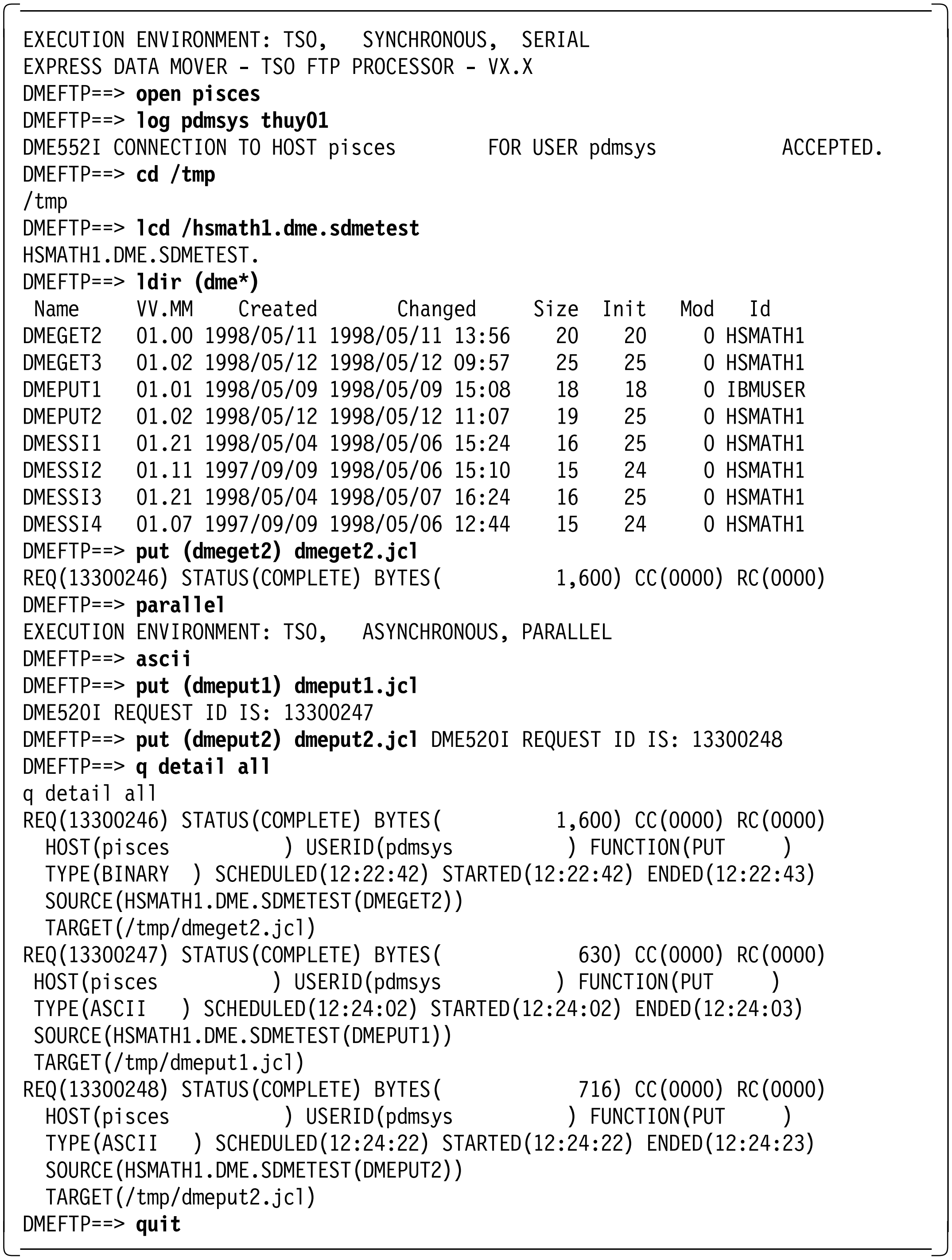
```
//DDJOB    JOB (USER,XXXX),MSGLEVEL=1
//S1       EXEC PGM=DMEFTP
//STEPLIB  DD  DSN=PDM.SDMELOAD,DISP=SHR
//SYSOUT   DD  SYSOUT=*
//SYSUT1   DD  *
* Open remote node named "gemini"
   open gemini
* Turn off echoing so password does not appear in output
   echo off
* Identify userid and password to "gemini"
   log user01 mypasswd
* Turn echoing back on to display commands
   echo on
   notify mymvsid
   lcd /mymvsid
   cd /bin
   putx mvsfile.tobe.copied dd                                         X
     op=/usr/user01/files/new.fil cbs=80 conv=block
   close
   end
//
```

Figure 119. Sample JCL For PDM MVS File Transfer Client To Read a MVS Data Set From a UNIX Application

12.2.2 Using The PDM Interactive TSO Client

12.2.2.1 Writing Data from MVS to UNIX Using the Interactive Client

If you are using PDM interactively in TSO, then Figure 120 shows how the dialog goes.



```
EXECUTION ENVIRONMENT: TSO,   SYNCHRONOUS,  SERIAL
EXPRESS DATA MOVER - TSO FTP PROCESSOR - VX.X
DMEFTP==> open pisces
DMEFTP==> log pdmsys thuy01
DME552I CONNECTION TO HOST pisces        FOR USER pdmsys           ACCEPTED.
DMEFTP==> cd /tmp
/tmp
DMEFTP==> lcd /hsmath1.dme.sdmetest
HSMATH1.DME.SDMETEST.
DMEFTP==> ldir (dme*)
 Name     VV.MM    Created       Changed     Size  Init   Mod   Id
DMEGET2   01.00 1998/05/11 1998/05/11 13:56    20    20     0 HSMATH1
DMEGET3   01.02 1998/05/12 1998/05/12 09:57    25    25     0 HSMATH1
DMEPUT1   01.01 1998/05/09 1998/05/09 15:08    18    18     0 IBMUSER
DMEPUT2   01.02 1998/05/12 1998/05/12 11:07    19    25     0 HSMATH1
DMESSI1   01.21 1998/05/04 1998/05/06 15:24    16    25     0 HSMATH1
DMESSI2   01.11 1997/09/09 1998/05/06 15:10    15    24     0 HSMATH1
DMESSI3   01.21 1998/05/04 1998/05/07 16:24    16    25     0 HSMATH1
DMESSI4   01.07 1997/09/09 1998/05/06 12:44    15    24     0 HSMATH1
DMEFTP==> put (dmeget2) dmeget2.jcl
REQ(13300246) STATUS(COMPLETE) BYTES(         1,600) CC(0000) RC(0000)
DMEFTP==> parallel
EXECUTION ENVIRONMENT: TSO,   ASYNCHRONOUS, PARALLEL
DMEFTP==> ascii
DMEFTP==> put (dmeput1) dmeput1.jcl
DME520I REQUEST ID IS: 13300247
DMEFTP==> put (dmeput2) dmeput2.jcl DME520I REQUEST ID IS: 13300248
DMEFTP==> q detail all
q detail all
REQ(13300246) STATUS(COMPLETE) BYTES(         1,600) CC(0000) RC(0000)
  HOST(pisces          ) USERID(pdmsys          ) FUNCTION(PUT     )
  TYPE(BINARY  ) SCHEDULED(12:22:42) STARTED(12:22:42) ENDED(12:22:43)
  SOURCE(HSMATH1.DME.SDMETEST(DMEGET2))
  TARGET(/tmp/dmeget2.jcl)
REQ(13300247) STATUS(COMPLETE) BYTES(           630) CC(0000) RC(0000)
 HOST(pisces          ) USERID(pdmsys          ) FUNCTION(PUT     )
 TYPE(ASCII   ) SCHEDULED(12:24:02) STARTED(12:24:02) ENDED(12:24:03)
 SOURCE(HSMATH1.DME.SDMETEST(DMEPUT1))
 TARGET(/tmp/dmeput1.jcl)
REQ(13300248) STATUS(COMPLETE) BYTES(           716) CC(0000) RC(0000)
  HOST(pisces          ) USERID(pdmsys          ) FUNCTION(PUT     )
  TYPE(ASCII   ) SCHEDULED(12:24:22) STARTED(12:24:22) ENDED(12:24:23)
  SOURCE(HSMATH1.DME.SDMETEST(DMEPUT2))
  TARGET(/tmp/dmeput2.jcl)
DMEFTP==> quit
```

Figure 120. Using the MVS ISPF Client Interactive Session

12.2.3 Using JCL and the SSI

These samples are based on subsystem interface, where MVS application can access data on remote node and by using JCL. Typical applications would be:

- Simultaneous unload/load of remote databases

  In the past database-unload utilities on the source system created flat files that were copied to the remote system using traditional FTP or other utilities. Then database-load utilities read the copied file and load the data into the target database.

  With IBM exPress Data Mover installed you can change the JCL for the database-unload utilities to avoid creating a flat file and instead to specify the load utility on the remote node. When the MVS job (database-unload utility) is started, and opens it′s output data set, IBM exPress Data Mover automatically starts the specified load utility at the remote site and the data from the MVS job is piped to the remote executable. In this way, the data never has to be stored on interim media.

- Application conversion

  You may have an existing application that runs as an MVS application. The output of this job is normally a flat file, which you need as input for a remote UNIX or Windows NT application that performs analysis of the data. You can direct the application to store the data directly on the remote media or feed it directly to a UNIX analysis program running on the remote host.

- Shared application data

  You may have applications that run both on MVS and UNIX or Windows NT hosts that need to share data. You can place the data on either machine and run the application against the data without having to move or replicate the data. Note that the SSI is required only if the data resides on remote hosts. If the data resides on MVS and the application on UNIX, then you can use the standard features of PDM to access the data (GETX/PUTX).

- Copy-files-between-UNIX-or-Windows-NT-servers

  By using MVS copy utilities, such as IEBGENER or IDCAMS, files can be copied from one remote system to another. On the input statement, specify which remote file is to copied and on the output statement, you specify which file is to be created on which remote system.

  The JCL for the MVS job to execute the MVS utility can be submitted from the MVS side or from either of the remote systems using the JES option of the PDM client.

- Piping data between applications running on different UNIX servers

  By using MVS copy utilities, such as IEBGENER or IDCAMS, data can be piped from an application on one remote system to an application on another remote system. On the input statement, specify the executable that creates the data and on the output statement, you specify which executable on which remote system should receive the data. The data is piped through MVS without being stored anywhere.

  The JCL for the MVS job to execute the MVS utility can be submitted from the MVS side or from either of the remote systems using the JES option of the PDM client.

12.2.3.1 Copying a UNIX or Windows NT File to MVS

This example in Figure 121 shows how a UNIX or Windows NT file can be copied to MVS using a utility. The utility used in this example, IEBGENER, copies data from the SYSUT1 dd statement to the SYSUT2 dd statement. Specifying DATATYPE=A, causes the IBM exPress Data Mover to convert the data from ASCII to EBCDIC before it is written into the MVS data set.

Note: DCB information must be specified for the input as the utility (in this case IEBGENER) or some other program does not get this information automatically, as it does when reading an MVS dataset.

```
//UXTOMVS  JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//*********************************************************************
//*        USE THE SSI AND IEBGENER TO COPY A FILE FROM UNIX          *
//*********************************************************************
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=′/pdmssitest/cobsamp4′,
//             SUBSYS=(DMES,′HOST=trnsun1m′,′USER=edmusera′,
//             ′PASS=edmusera′,′FILETYPE=F′,′DATATYPE=A′),
//             DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSUT2   DD  DSN=DME110.TRAIN.COBS1,DISP=(,CATLG,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//             SPACE=(CYL,(10,10),RLSE),UNIT=3390,VOL=SER=TEST01
//SYSIN    DD  DUMMY
//
```

Figure 121. SSI-JCL to Copy a UNIX or Windows NT File to MVS

12.2.3.2 Copying MVS Data Sets to UNIX or Windows NT

In the example shown in Figure 122, an MVS utility is used to do two things:

1. Copy the data from the MVS file PDM.XDSN1 into file */pdmssitest/odsnx* on a remote system.

2. Copy the data from the MVS file PDMXDSN2 and append it to the file */pdmssitest/odsnx* on a remote system.

Specifying DATATYPE=N causes the IBM exPress Data Mover to convert the data from EBCDIC to ASCII without truncating trailing blanks before it is transmitted to the remote system.

```
//MVSTOUX JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//**********************************************************************
//*        USE THE SSI W/IDCAMS TO COPY DATASETS TO UNIX               *
//**********************************************************************
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//MVSIN1   DD  DSN=PDM.XDSN1,DISP=SHR
//MVSIN2   DD  DSN=PDM.XDSN2,DISP=SHR
//UNIXOUT1 DD  DSN=′/pdmssitest/odsnx′,
//             DCB=(RECFM=F,LRECL=133),
//             SUBSYS=(DMES,′HOST=trnsun1m′,′USER=edmusera′,
//             ′PASS=edmusera′,′FILETYPE=F′,′DATATYPE=N′)
//UNIXOUT2 DD  DSN=′/pdmssitest/odsnx′,
//             DCB=(RECFM=F,LRECL=133),
//             SUBSYS=(DMES,′HOST=trnsun1m′,′USER=edmusera′,
//             ′PASS=edmusera′,′FILETYPE=F′,′DATATYPE=N′,
//             ′FILEDISP=MOD′)
//SYSIN    DD  *
  REPRO INFILE(MVSIN1) OUTFILE(UNIXOUT1)
  REPRO INFILE(MVSIN2) OUTFILE(UNIXOUT2)
/*
//
```

Figure 122. SSI-JCL to Copy MVS Data Sets to UNIX or Windows NT

12.2.3.3 Reading a UNIX or Windows NT File from an MVS Application

In order to read data directly from a UNIX or Windows NT file into an MVS application the input dd statement should identify the file to be read. This is basically the same as described in 12.2.3.1, “Copying a UNIX or Windows NT File to MVS” on page 151. Figure 123 shows some skeleton JCL statements to read a file from a remote system.

Note: DCB must be specified for the input file if not hard coded in the application

```
//*       ...
//STEP    EXEC PGM=ANYPRGM
//*       ...
//INPUT    DD  SUBSYS=(DMES,′HOST=........′,′PROFILE=.....′,
//             ′PATH=/..../...../.....′,
//             ′FILETYPE=F′,′DATATYPE=..′),
//             DCB=(RECFM=...,LRECL=...,BLKSIZE=...)
//*       ...
```

Figure 123. SSI-JCL to Read a UNIX or Windows NT File

12.2.3.4 Writing a UNIX or Windows NT File from an MVS Application

In order to write data directly to a UNIX or Windows NT file from an MVS application the output dd statement should identify the file to be created. Figure 124 shows some skeleton JCL statements to create a file on a remote system.

```
//*       ...
//STEP    EXEC PGM=ANYPRGM
//*       ...
//OUTPUT   DD  SUBSYS=(DMES,′HOST=........′,′PROFILE=.....′,
//             ′PATH=/..../...../.....′,
//             ′FILETYPE=F′,′DATATYPE=..′),DCB=...
//*       ...
```

Figure 124. SSI-JCL to Write a UNIX or Windows NT File

Note: DCB must be specified for the output file if not hard coded in the application.

12.2.3.5 Piping Data from a UNIX Executable

Figure 125 shows how data is piped from the UNIX utility *tar* to MVS initiated using JCL. The FILETYPE must be X and you need to specify the executable using the PATH subparameter. The input file that *tar* reads is defined in the arguments. This sample performs the same task as shown in Figure 118 on page 147.

```
//UXPTMVS  JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//**********************************************************************
//*        PIPE DATA FROM TAR TO IEBGENER                              *
//**********************************************************************
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  SUBSYS=(DMES,′HOST=gemini′,′USER=user01′,
//             ′PASS=mypasswd′,′FILETYPE=X′,′DATATYPE=B′,
//             ′PATH=/etc/sbin/tar′,
//             ′FILEARGS=-xrv /usr/user01/files/source.fil′),
//             DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//SYSUT2   DD  DSN=MYMVSID.SOURCE.FIL.ARCHIVE(+1),DISP=(,CATLG,DELETE),
//             DCB=*.SYSUT1,,UNIT=3490
//SYSIN    DD  DUMMY
//
```

Figure 125. SSI-JCL to Pipe Data from a UNIX Executable

12.2.3.6 Piping Data to a UNIX Executable

Figure 126 shows how data is piped to the UNIX utility *dd* from MVS, initiating the piping using JCL. The FILETYPE must be X and you need to specify the executable using the PATH subparameter. The output file *dd* creates is defined in the arguments. This sample performs the same task as shown in Figure 119 on page 148.

```
//MVSPTUX  JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//**********************************************************************
//*        PIPE DATA FROM IEBGENER TO DD                               *
//**********************************************************************
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN=MYMVSID.MVSFILE.TOBE.COPIED,DISP=SHR
//SYSUT2   DD  SUBSYS=(DMES,′HOST=gemini′,′USER=user01′,
//             ′PASS=mypasswd′,′FILETYPE=X′,′DATATYPE=B′,
//             ′PATH=/bin/dd′,
//         ′FILEARGS=op=/usr/user01/files/new.fil cbs=80 conv=block),
//             DCB=MYMVSID.MVSFILE.TOBE.COPIED
//SYSIN    DD  DUMMY
//
```

Figure 126. SSI-JCL to Pipe Data to a UNIX Executable

12.2.3.7 Piping Data Between an MVS Application and a UNIX Executable

Figure 127 shows some skeleton JCL statement to pipe data between a MVS application and UNIX executables.

Note: DCB must be specified for the files if not hard coded in the application.

```
//*        ...
//STEP     EXEC PGM=ANYPRGM
//*        ...
//INPUT     DD  SUBSYS=(DMES,'HOST=........','PROFILE=.....',
//              'PATH=/..../...../..anyexec..',
//              'FILETYPE=X),DCB=...
//*        ...
//OUTPUT    DD  SUBSYS=(DMES,'HOST=........','PROFILE=.....',
//              'PATH=/..../...../..anyexec..',
//              'FILETYPE=X),DCB=...
//*        ...
```

Figure 127. SSI-JCL to Pipe Data Between an MVS Application and UNIX Executables

In Figure 127, the USER, PASS, and DATATYPE subparameters are not used. Instead the PROFILE parameter has been specified to define a name of a profile containing that information (see Figure 128). This profile is stored in the the profile data set. The profile data set is a PDS and the named *userid.DME.PROFILE*. The *userid* is the userid for this jobstep is running under. This makes it easier and more secure to hold user related information for the open system (user id, password).

```
* SAMPLE PROFILE
USER     pdmsys
PASS     thuy01
DATATYPE A
NOTIFY   hsmath1
```

Figure 128. SSI PROFILE Sample

12.2.3.8 Copying Data Between UNIX or Windows NT Disks via MVS

Figure 129 on page 156 shows how data can be copied from one remote system to another via MVS. Such copy jobs can be initiated from:

- MVS
- Sending UNIX or Windows NT
- Receiving UNIX or Windows NT

The JCL shown in Figure 129 on page 156 initializes the copy independent from where the activation happens. When initiating the copy process from MVS, just submit the job.

```
//UNIXUNIX JOB (001),'SSI TEST',CLASS=A,MSGCLASS=Z,REGION=0M
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  DSN='/pdmssitest/cobsamp4',
//             DCB=(RECFM=F,BLKSIZE=32760),
//             SUBSYS=(DMES,'HOST=trnsun1m','USER=edmusera',
//             'PASS=edmusera','FILETYPE=F','DATATYPE=B')
//SYSUT2   DD  DSN='pdmssitest/cobsamp4',
//             DCB=*.SYSUT1,
//             SUBSYS=(DMES,'HOST=trnsun2m','USER=edmuserb',
//             'PASS=edmuserb','FILETYPE=F','DATATYPE=B')
//SYSIN    DD  DUMMY
//
```

Figure 129. SSI-JCL to Copy UNIX or Windows NT to UNIX or Windows NT

If the copy process is to be initiated from either the sending or receiving system, then a file on that system must contain the JCL, as shown in the sample in Figure 129. When starting the copy process from the UNIX or Windows NT system, set the DMEFTP program into JES mode and submit (put) the file that contains the JCL. Figure 130 shows how to set DMEFTP to JES mode. The PUT statement submits the JCL file to the MVS server. The Q DETAIL REQID statement gives to the user the request ID. The GET statement returns the job output to the user and contains normal job output information.

When submitting JCL from UNIX, the JOB statement must not contain a USER= keyword because PDM supplies this value.

```
UNIX DMEFTP commands:

>dmeftp
>open 3090                          <=== MVS host id
>mvs ibmuser ibm                    <=== MVS user name and pw
>acct pdmsys pdmsyspw               <=== node user name and pw
>jes                                <=== set DMEFTP to JES mode
>put /pdmtest/jclfile               <=== file that contains the JCL
>q detail reqid                     <=== get the requestid
>dir                                <=== get JES JOBID
>get
>seq                                <=== back to sequential
```

Figure 130. Submit MVS-JCL from a PDM Client

Note that even though PDM does not directly support OpenEdition (MVS UNIX) files, the JCL shown in Figure 129 should support it if you change the SYSUT2 DD statement to the code shown in Figure 131.

```
//SYSUT2   DD PATH='.......................',    <--- HFS file name
//            PATHDISP=(KEEP,DELETE),
//            PATHMODE=(SIRUSR,SIWUSR),          <--- Only user can use
//            PATHOPTS=OCREAT,                   <--- new file
//            DCB=*.SYSUT1
```

Figure 131. SSI-JCL For OpenEdition

This use of PATH requires DFSMS/MVS 1.3 with a PTF (or a later version) to allow BSAM and QSAM to support HFS.

12.2.3.9 Piping Data Between UNIX Applications in Different Nodes

Figure 132 shows some skeleton JCL statement to pipe data from one UNIX executable in one UNIX system to another UNIX executable in a different UNIX system. The JCL can be submitted in the same way as described in 12.2.3.8, “Copying Data Between UNIX or Windows NT Disks via MVS” on page 155.

Note: DCB must be specified even if not used by UNIX

```
//UXUXUXUX JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//*********************************************************************
//*        PIPE DATA FROM UNIX TO UNIX                                *
//*********************************************************************
//STEP    EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  SUBSYS=(DMES,′HOST=..from......′,′PROFILE=.....′,
//             ′PATH=/..../...../..oneexec..′,
//             ′FILETYPE=X),
//             DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//SYSUT2   DD  SUBSYS=(DMES,′HOST=..to......′,′PROFILE=.....′,
//             ′PATH=/..../...../..anotherexec..′,
//             ′FILETYPE=X),DCB=*.SYSUT1
//SYSIN    DD  DUMMY
```

Figure 132. SSI-JCL to Pipe Data Between UNIX Executables

12.3 Backing up Data to S/390 Environment

When the IBM InfoSpeed solution was announced, a Statement of Direction (SOD) was issued for ADSM stating that ADSM will be able to use IBM exPress Data Mover to send data. Before this support is available (and later as a complement) IBM exPress Data Mover can be used to do full-volume backup or to back up selected files or databases.

IBM exPress Data Mover can also be used to reduce the need for local tapes on UNIX or Windows NT servers. Files can be copied to physical or virtual tapes or tape libraries attached to OS/390 using IBM exPress Data Mover.

12.3.1 Backing up Databases from UNIX or Windows NT to OS/390

One of the biggest challenges in availability management today is how to back up large ERP production data bases from UNIX environment with no or minimal interrupt to the online service. Several ERP (SAP, BAAN, Lotus Notes, production environments need almost continuous availability and a reliable secure environment. One option, illustrated in Figure 133 on page 158, is to use InfoSpeed to copy the production databases and archive logs to the S/390 attached tape library, such as the IBM 3494 Virtual-Tape-Server. This offers automation, management (through Removable Media Manager (RMM)), and reliable media to the open environment. It provides good investment protection by sharing the tape library between different platforms.

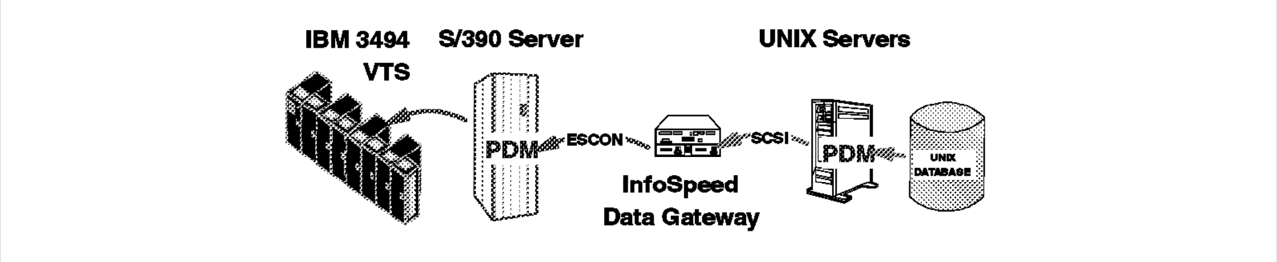


Figure 133. Backing up UNIX/NT Data Base to S/390 Tape Library

12.3.2 Backing up Data from UNIX to S/390 Environment

As interim backup solutions until ADSM is available, you can use:

- UNIX utilities such as *tar*. Although tar (*t*ape *ar*chive) was designed to go to tape, you can direct the output to a named or unnamed pipe. PDM supports either one to allow the tar output to go via InfoSpeed directly to an MVS sequential disk data set or an MVS tape or tape library.

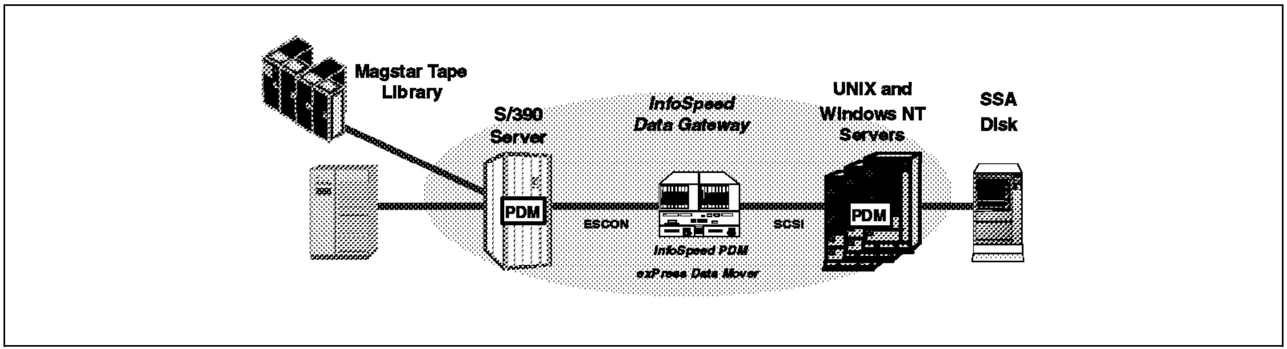


Figure 134. Backing up Data from UNIX to S/390 Environment

The following steps make up one possible approach:

- In a UNIX script define a named pipe.
- In the script, invoke tar as a separate process, telling it which files to back up and the name of the pipe to write the output to.
- In the script, invoke PDM to submit an MVS job that reads from the UNIX named pipe and writes to MVS disk or tape.

The script can monitor and record the name of the MVS data set and what it contains, in order to do a later restore. The job may look like the example shown in Figure 135 on page 159.

```
//UXMVSPDM JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//         EXEC PGM=DMEFTP,PARM=′SOURCE(HSMAT3)′  (my profile)
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
* Host, userID and password are in my profile.
binary
* ″unixfile″ management class is assumed to cause quick migration.
lsite mgmtclas unixfile
* Set SPACE=(CYL,(30,10)),RECFM=V,LRECL=32756
lsite cylinders
lsite primary 50
lsite secondary 30
lsite recfm v
lsite lrecl 32756
* Copy from tar′s named output pipe to MVS tape.
get /usr/payroll/weekly/tar PAYROLL.WEEKLY.TAR.D9837
/*
```

Figure 135. Example of JCL to Back up UNIX Data Using tar

There are many database backup programs that are designed to go to tape or standard output and they should work exactly like tar here.

- PDM, which directly copies the file without using *tar* as shown in 12.2.1.1, “Copying a UNIX file to MVS (GET)” on page 142, 12.2.1.3, “Copying a UNIX File To a MVS Tape Data Set” on page 144, or 12.2.3.1, “Copying a UNIX or Windows NT File to MVS” on page 151.
- Using UNIX utilities to do volume backup. The JCL in Figure 136 shows how to back up a raw partition in an HP environment. An alternative is to have the *dd* read a UNIX logical volume and pipe the output through PDM to an MVS data set.

```
//HPRAW    JOB (001),′SSI TEST′,CLASS=A,MSGCLASS=Z,REGION=0M
//*********************************************************************
//*        BACK UP HP RAW PARTITION                                   *
//*********************************************************************
//STEP    EXEC PGM=IEBGENER
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  SUBSYS=(DMES,′HOST=hpsys′,′PROFILE=kjellen′,
//             ′PATH=/dev/rdsk/c0t0d3′,
//             ′FILETYPE=F,′DATATYPE=B′),
//             DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//SYSUT2   DD  UNIT=TAPE,DISP=(,CATLG,DELETE),
//             DSN=HP.RAW.PART.C0T0D3.BACKUP(+1),
//             DCB=*.SYSUT1
//SYSIN    DD  DUMMY
```

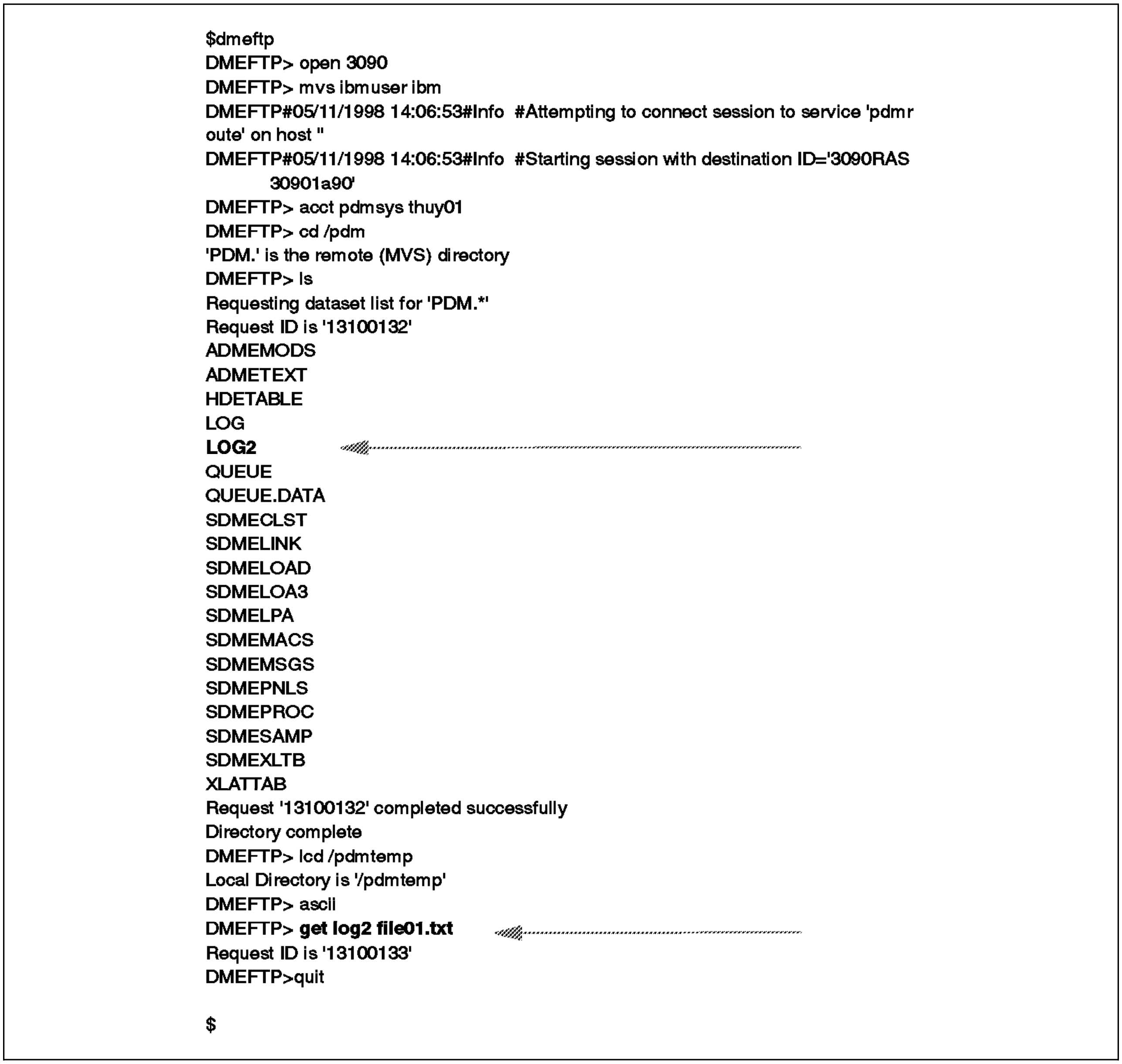
Figure 136. SSI-JCL to Back up an HP Raw Partition

12.4 Data-Copy Sharing Samples from the UNIX or Windows NT

An alternative to submitting batch jobs in MVS to copy data to or from a UNIX or Windows NT environment is to use the PDM Client in UNIX or Windows NT.

12.4.1 Get Data from MVS

Figure 137 illustrates how a MVS data set PDM.LOG2 is copied to a file /pdmtemp/file01.txt in UNIX.

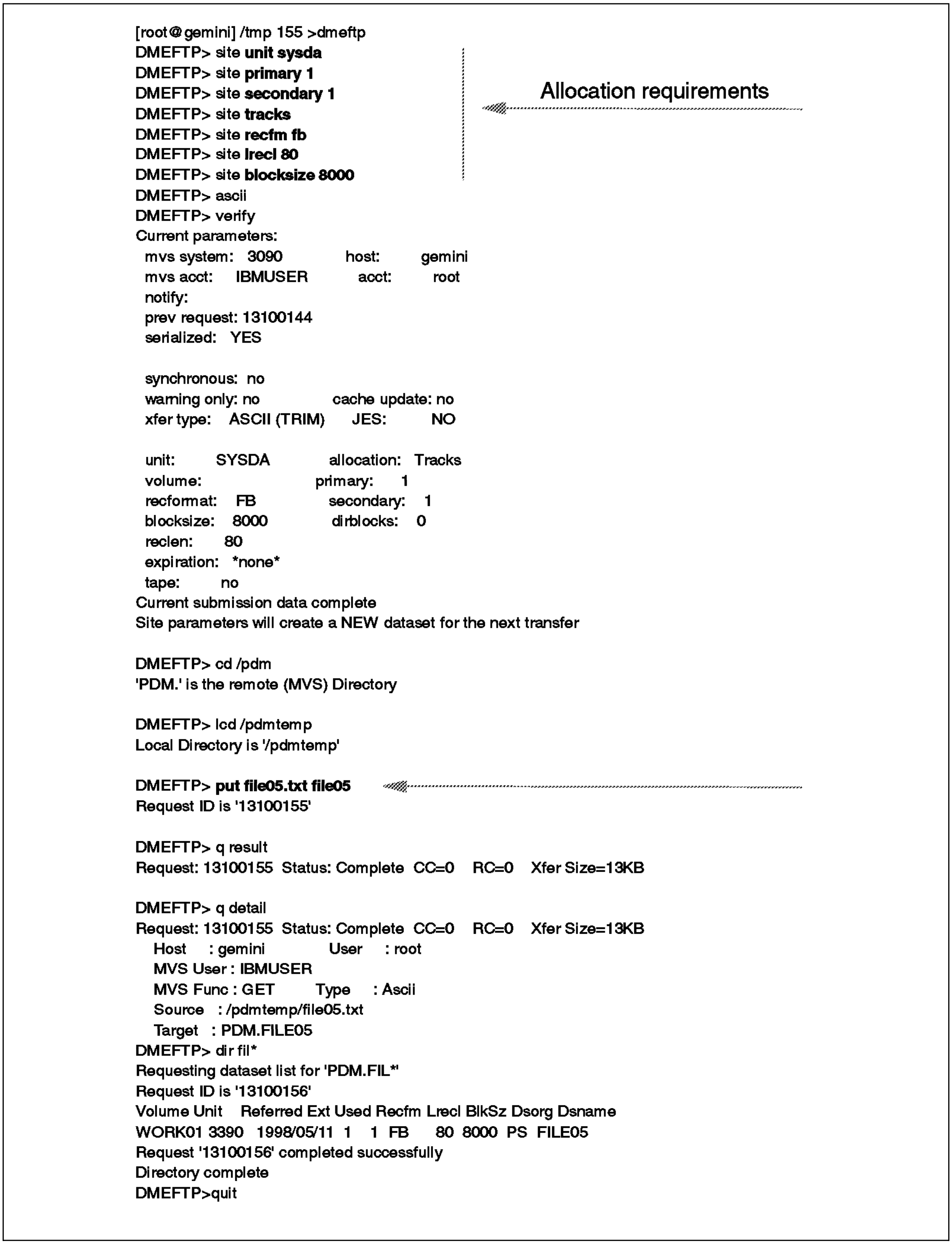


```
$dmeftp
DMEFTP> open 3090
DMEFTP> mvs ibmuser ibm
DMEFTP#05/11/1998 14:06:53#Info  #Attempting to connect session to service 'pdmr
oute' on host ''
DMEFTP#05/11/1998 14:06:53#Info  #Starting session with destination ID='3090RAS
          30901a90'
DMEFTP> acct pdmsys thuy01
DMEFTP> cd /pdm
'PDM.' is the remote (MVS) directory
DMEFTP> ls
Requesting dataset list for 'PDM.*'
Request ID is '13100132'
ADMEMODS
ADMETEXT
HDETABLE
LOG
LOG2
QUEUE
QUEUE.DATA
SDMECLST
SDMELINK
SDMELOAD
SDMELOA3
SDMELPA
SDMEMACS
SDMEMSGS
SDMEPNLS
SDMEPROC
SDMESAMP
SDMEXLTB
XLATTAB
Request '13100132' completed successfully
Directory complete
DMEFTP> lcd /pdmtemp
Local Directory is '/pdmtemp'
DMEFTP> ascii
DMEFTP> get log2 file01.txt
Request ID is '13100133'
DMEFTP>quit

$
```

Figure 137. Sample Script to Get Data from MVS

12.4.2 Put Data on MVS

This operation is a little more complicated since MVS does not allow the transfer of data to its devices without some information about the kind of allocation required. You have to specify space requirements, the type of data set you intend to create, the record length, the format, and so on. Figure 138 on page 162 shows an example of copying data from /pdmtemp/file05.txt to MVS sequential data set PDM.FILE05 in fixed block format.

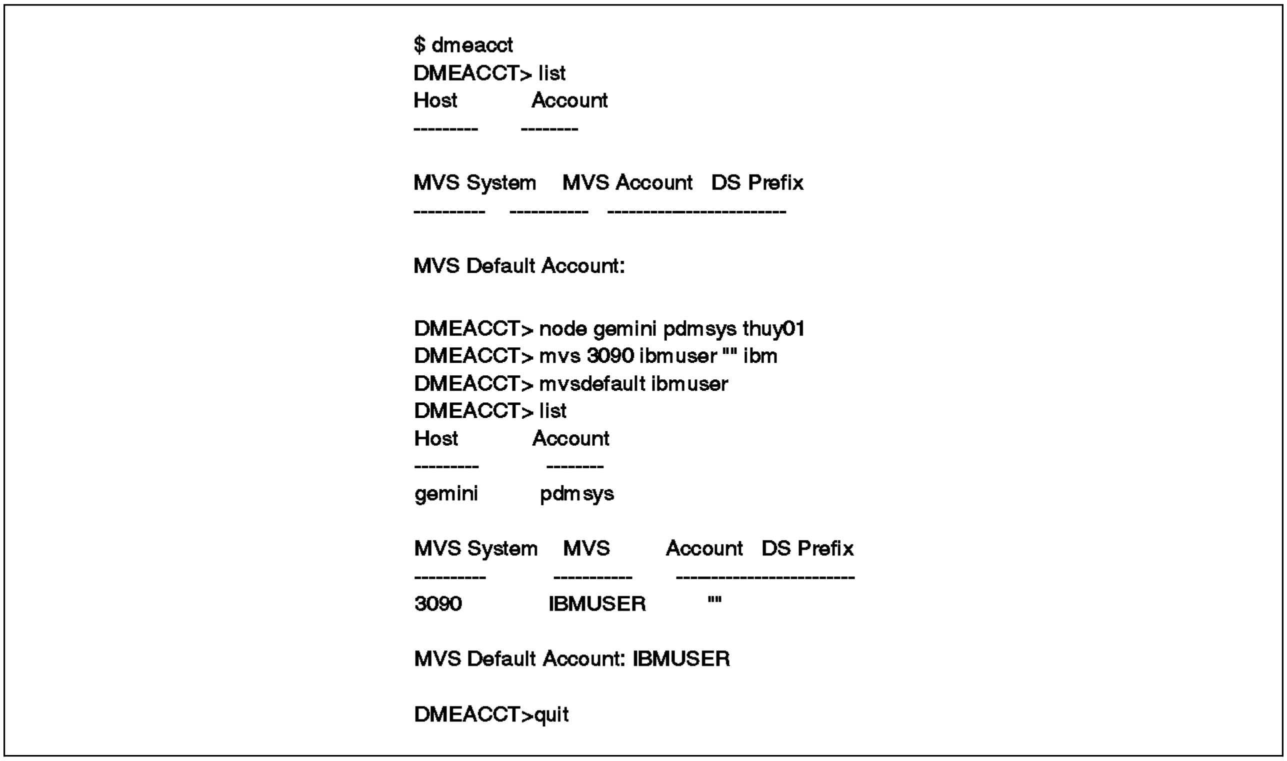


```
[root@gemini] /tmp 155 >dmeftp
DMEFTP> site unit sysda
DMEFTP> site primary 1
DMEFTP> site secondary 1
DMEFTP> site tracks
DMEFTP> site recfm fb
DMEFTP> site lrecl 80
DMEFTP> site blocksize 8000
DMEFTP> ascii
DMEFTP> verify
Current parameters:
  mvs system:   3090           host:       gemini
  mvs acct:   IBMUSER         acct:       root
  notify:
  prev request: 13100144
  serialized:   YES

  synchronous:  no
  warning only: no              cache update: no
  xfer type:    ASCII (TRIM)     JES:        NO

  unit:        SYSDA            allocation:   Tracks
  volume:                      primary:     1
  recformat:    FB              secondary:    1
  blocksize:   8000             dirblocks:    0
  reclen:      80
  expiration:   *none*
  tape:         no
Current submission data complete
Site parameters will create a NEW dataset for the next transfer

DMEFTP> cd /pdm
'PDM.' is the remote (MVS) Directory

DMEFTP> lcd /pdmtemp
Local Directory is '/pdmtemp'

DMEFTP> put file05.txt file05
Request ID is '13100155'

DMEFTP> q result
Request: 13100155  Status: Complete  CC=0    RC=0    Xfer Size=13KB

DMEFTP> q detail
Request: 13100155  Status: Complete  CC=0    RC=0    Xfer Size=13KB
    Host     : gemini             User     : root
    MVS User : IBMUSER
    MVS Func : GET         Type     : Ascii
    Source    : /pdmtemp/file05.txt
    Target   : PDM.FILE05
DMEFTP> dir fil*
Requesting dataset list for 'PDM.FIL*'
Request ID is '13100156'
Volume Unit    Referred Ext Used Recfm Lrecl BlkSz Dsorg Dsname
WORK01 3390   1998/05/11  1    1  FB      80  8000  PS  FILE05
Request '13100156' completed successfully
Directory complete
DMEFTP>quit
```

Figure 138. Sample Script to Put Data on MVS

12.4.3 Set up Default Account Information

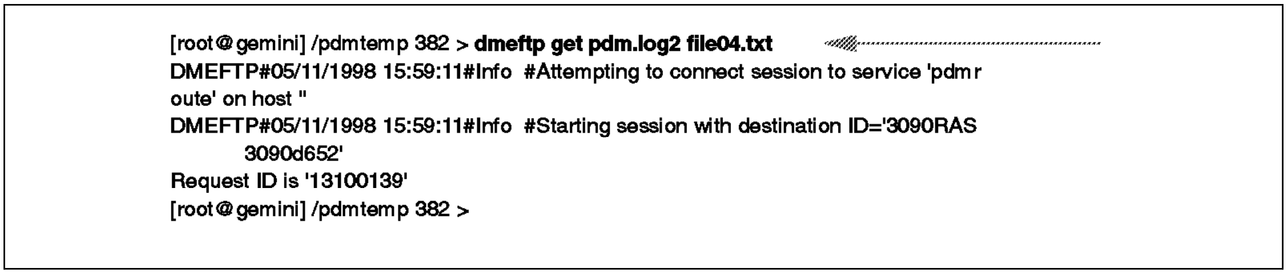
Every time you need to access data on MVS, you need to provide accounting information. The procedure can be simplified by setting up a default account information file, as shown in Figure 139.



```
$ dmeacct
DMEACCT> list
Host          Account
---------     --------

MVS System    MVS Account   DS Prefix
----------    -----------   ------------------------

MVS Default Account:


DMEACCT> node gemini pdmsys thuy01
DMEACCT> mvs 3090 ibmuser "" ibm
DMEACCT> mvsdefault ibmuser
DMEACCT> list
Host          Account
---------     --------
gemini        pdmsys

MVS System    MVS         Account   DS Prefix
----------    -----------     -------------------------
3090          IBMUSER         ""

MVS Default Account: IBMUSER

DMEACCT>quit
```

Figure 139. Script to Set up Default Account Information in PDM

With default account information, copying the MVS data set PDM.LOG2 is reduced to a single command, as in Figure 140.



```
[root@gemini] /pdmtemp 382 > dmeftp get pdm.log2 file04.txt
DMEFTP#05/11/1998 15:59:11#Info  #Attempting to connect session to service 'pdmr
oute' on host ''
DMEFTP#05/11/1998 15:59:11#Info  #Starting session with destination ID='3090RAS
        3090d652'
Request ID is '13100139'
[root@gemini] /pdmtemp 382 >
```

Figure 140. Script Defining a Source File for PDM

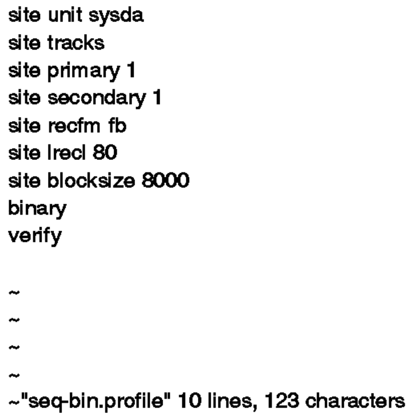
12.4.4 Using Default Allocation Information on OS/390

Every time you need to put data on MVS and a DFSMS data class has not been defined, you need to specify allocation information. The procedure can be simplified by setting up various default allocation files, also called *source files*:

1. Open a file using vi:

```
[root@gemini] / 128 > cd /u/pdmsys
[root@gemini] /u/pdmsys 129 > vi seq-bin.profile
```

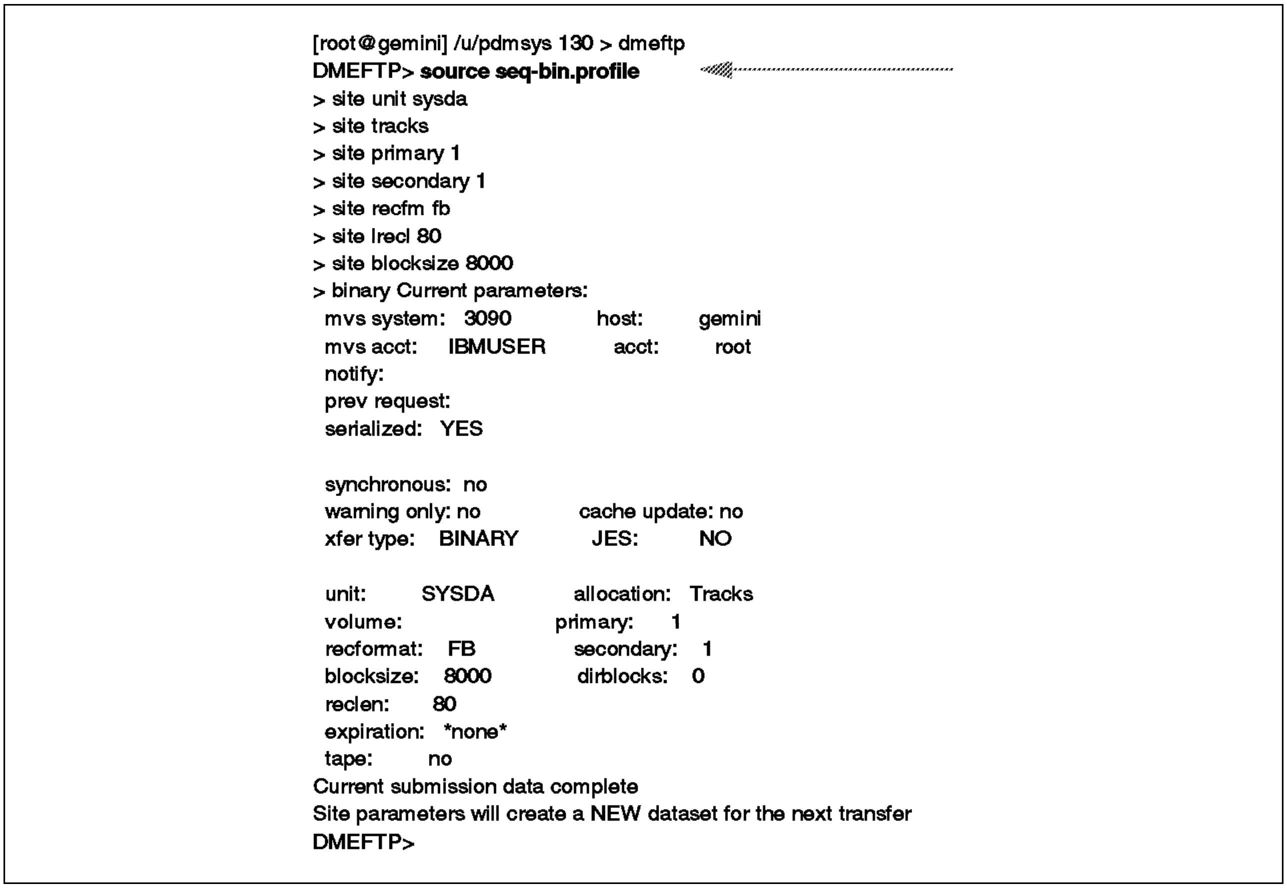
2. Fill in the file with the following parameters:



```
site unit sysda
site tracks
site primary 1
site secondary 1
site recfm fb
site lrecl 80
site blocksize 8000
binary
verify

~
~
~
~
~"seq-bin.profile" 10 lines, 123 characters
```

3. Save the file (*:wq!* command)

4. Start dmeftp, as shown in Figure 141.

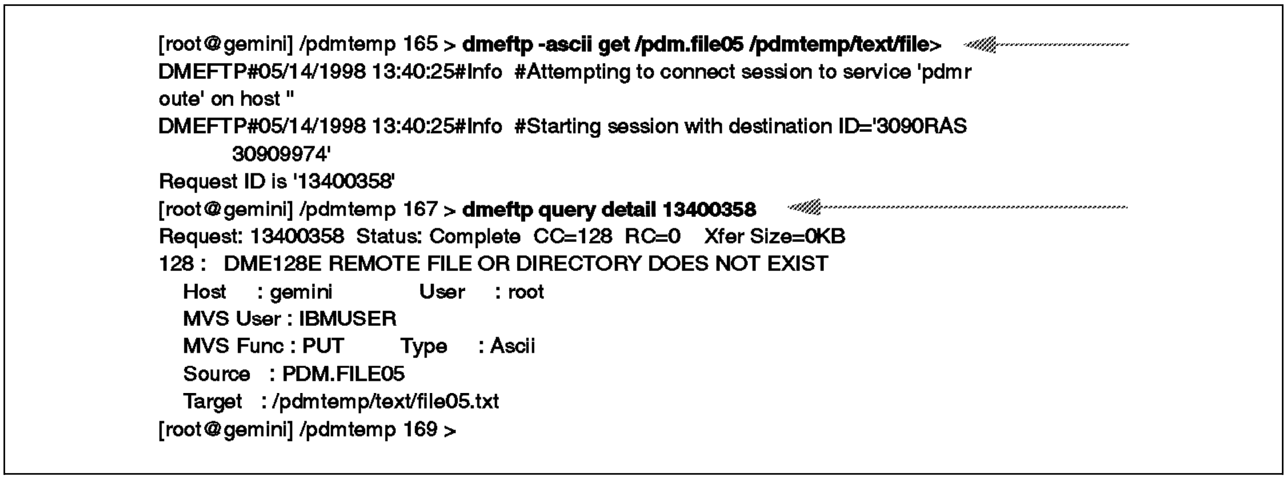


```
[root@gemini] /u/pdmsys 130 > dmeftp
DMEFTP> source seq-bin.profile
> site unit sysda
> site tracks
> site primary 1
> site secondary 1
> site recfm fb
> site lrecl 80
> site blocksize 8000
> binary Current parameters:
  mvs system:   3090          host:       gemini
  mvs acct:     IBMUSER         acct:       root
  notify:
  prev request:
  serialized:   YES

  synchronous:  no
  warning only: no            cache update: no
  xfer type:    BINARY          JES:        NO

  unit:         SYSDA         allocation:   Tracks
  volume:                    primary:      1
  recformat:    FB             secondary:    1
  blocksize:    8000           dirblocks:    0
  reclen:       80
  expiration:   *none*
  tape:         no
Current submission data complete
Site parameters will create a NEW dataset for the next transfer
DMEFTP>
```

Figure 141. Script to Start DMEFTP Using a Profile Source File

12.4.5 Submitting an Asynchronous Request

When operating asynchronously, the results of a request are not discovered until the request has completed. Figure 142 on page 165 is an example script for the submission of an asynchronous request that will fail, followed by a status check of the request at a later time:



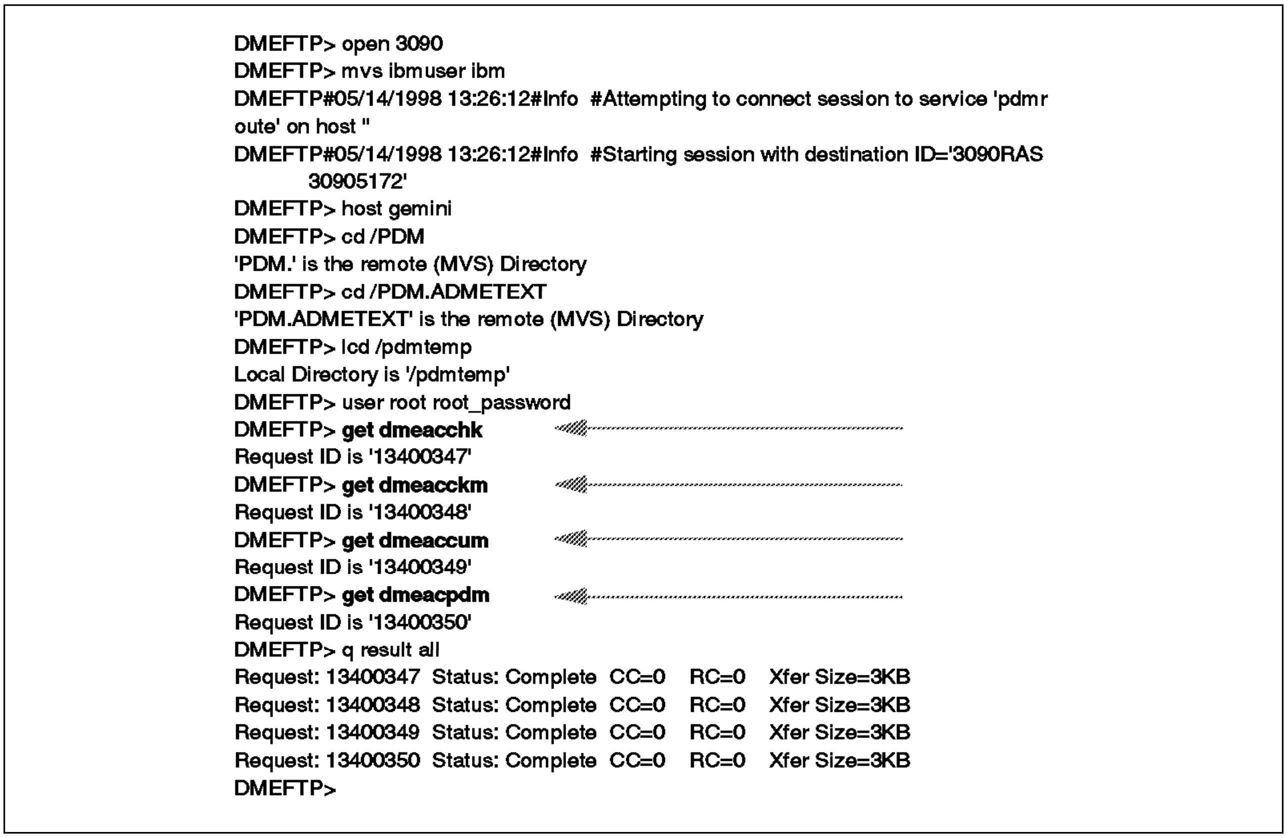
```
[root@gemini] /pdmtemp 165 > dmeftp -ascii get /pdm.file05 /pdmtemp/text/file>
DMEFTP#05/14/1998 13:40:25#Info  #Attempting to connect session to service 'pdmr
oute' on host ''
DMEFTP#05/14/1998 13:40:25#Info  #Starting session with destination ID='3090RAS
        30909974'
Request ID is '13400358'
[root@gemini] /pdmtemp 167 > dmeftp query detail 13400358
Request: 13400358  Status: Complete  CC=128  RC=0   Xfer Size=0KB
128 :   DME128E REMOTE FILE OR DIRECTORY DOES NOT EXIST
   Host     : gemini            User     : root
   MVS User : IBMUSER
   MVS Func : PUT        Type     : Ascii
   Source   : PDM.FILE05
   Target   : /pdmtemp/text/file05.txt
[root@gemini] /pdmtemp 169 >
```

Figure 142. Exampl Script for an Asynchronous Request

12.4.6 Copying Several Binary Files

Figure 143 is an example of copying several binary files from MVS to UNIX using the PDM Client.

Note: The default data set prefixes on MVS are used, and nonsynchronous and parallel operation is specified:

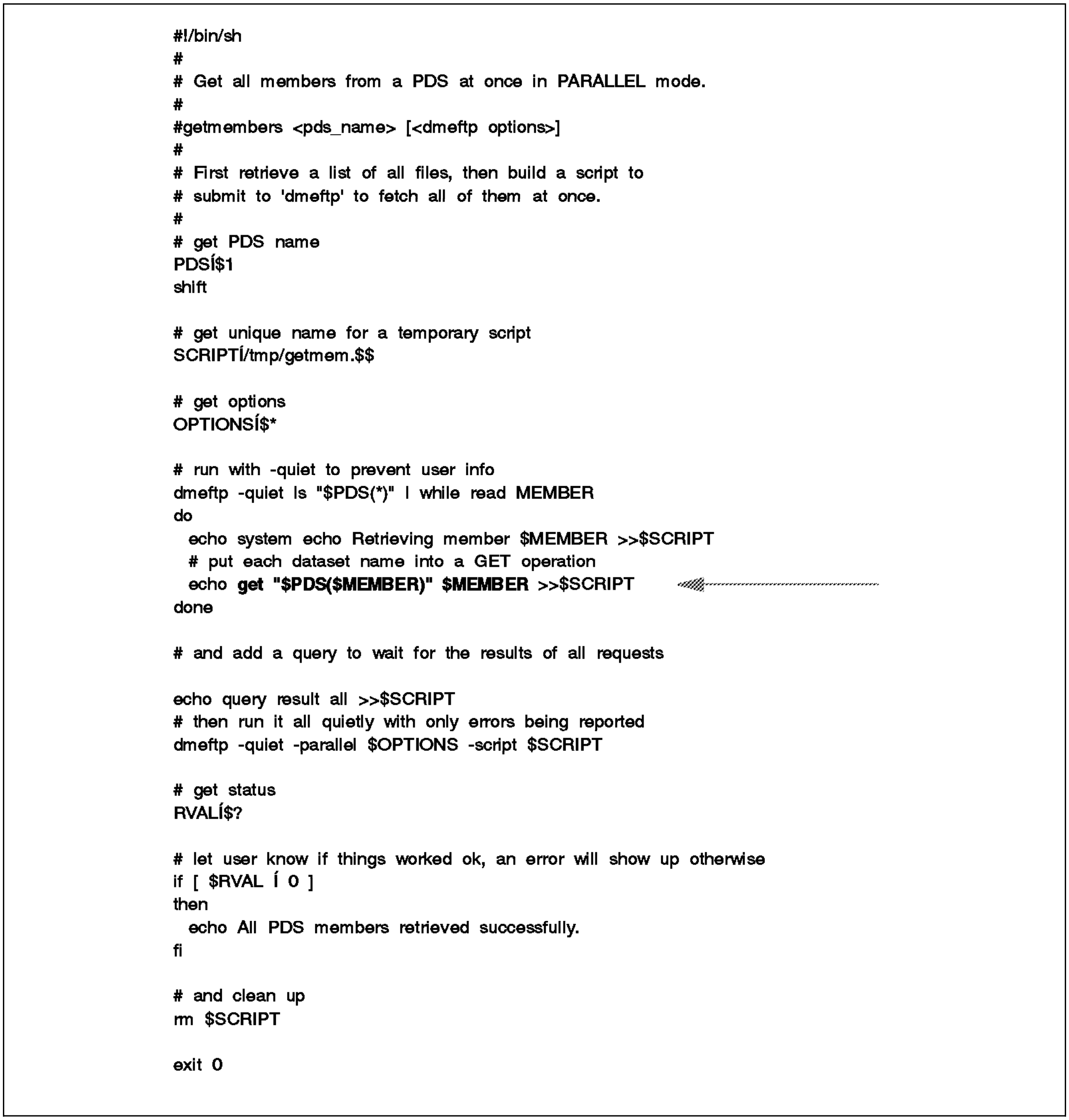


```
DMEFTP> open 3090
DMEFTP> mvs ibmuser ibm
DMEFTP#05/14/1998 13:26:12#Info  #Attempting to connect session to service 'pdmr
oute' on host ''
DMEFTP#05/14/1998 13:26:12#Info  #Starting session with destination ID='3090RAS
        30905172'
DMEFTP> host gemini
DMEFTP> cd /PDM
'PDM.' is the remote (MVS) Directory
DMEFTP> cd /PDM.ADMETEXT
'PDM.ADMETEXT' is the remote (MVS) Directory
DMEFTP> lcd /pdmtemp
Local Directory is '/pdmtemp'
DMEFTP> user root root_password
DMEFTP> get dmeacchk
Request ID is '13400347'
DMEFTP> get dmeacckm
Request ID is '13400348'
DMEFTP> get dmeaccum
Request ID is '13400349'
DMEFTP> get dmeacpdm
Request ID is '13400350'
DMEFTP> q result all
Request: 13400347  Status: Complete  CC=0   RC=0   Xfer Size=3KB
Request: 13400348  Status: Complete  CC=0   RC=0   Xfer Size=3KB
Request: 13400349  Status: Complete  CC=0   RC=0   Xfer Size=3KB
Request: 13400350  Status: Complete  CC=0   RC=0   Xfer Size=3KB
DMEFTP>
```

Figure 143. Example of Script to Copy Several Binary Files

12.4.7 Submitting Multiple Requests

Unlike traditional TCP/IP FTP, the PDM system can submit multiple requests for simultaneous transfer. For small files, this savings is not significant, but for larger amounts of data, parallel transfer of files can result in significant savings in overall throughput, because multiple channel paths are used. As in the previous example, Figure 143 on page 165, the example shown in Figure 144 fetches all of the members of a PDS in parallel mode.



```
#!/bin/sh
#
# Get all members from a PDS at once in PARALLEL mode.
#
#getmembers <pds_name> [<dmeftp options>]
#
# First retrieve a list of all files, then build a script to
# submit to 'dmeftp' to fetch all of them at once.
#
# get PDS name
PDSÍ$1
shift

# get unique name for a temporary script
SCRIPTÍ/tmp/getmem.$$

# get options
OPTIONSÍ$*

# run with -quiet to prevent user info
dmeftp -quiet ls "$PDS(*)" | while read MEMBER
do
  echo system echo Retrieving member $MEMBER >>$SCRIPT
  # put each dataset name into a GET operation
  echo get "$PDS($MEMBER)" $MEMBER >>$SCRIPT
done

# and add a query to wait for the results of all requests

echo query result all >>$SCRIPT
# then run it all quietly with only errors being reported
dmeftp -quiet -parallel $OPTIONS -script $SCRIPT

# get status
RVALÍ$?

# let user know if things worked ok, an error will show up otherwise
if [ $RVAL Í 0 ]
then
  echo All PDS members retrieved successfully.
fi

# and clean up
rm $SCRIPT

exit 0
```

Figure 144. Sequence of Instructions to Submit Multiple Requests

Appendix A. Configuration Worksheets

This appendix contains a chart to be used to configure an IBM InfoSpeed solution and matrixes to be used to document the configuration of direct ESCON-attached server and the configuration of an IBM InfoSpeed Data Gateway with one or two attached IBM InfoSpeed Multiplexors.

A.1 InfoSpeed Gateway Configuration

The normal procedure is that the IBM InfoSpeed Data Gateway is configured at manufacturing time. The procedure is to fax configuration matrixes to the factory. This procedure is described in detail in Chapter 5, "Install and Configure IBM InfoSpeed" on page 31 and in A.1.2, "Gateway Configuration Worksheets" on page 171.

A.1.1 Gateway Configuration Chart

The chart shown in Figure 145 on page 170 will be helpful in planning and sizing an IBM InfoSpeed solution where the IBM InfoSpeed Data Gateway and the IBM InfoSpeed Multiplexors are included. The chart can also be used for direct ESCON-attached configurations. The chart has the following parts already drawn:

A On the chart you find three figures representing one OS/390 image each. In these figures you should enter the SMF-ID specified in SYS1.PARMLIB for each OS/390 image, for example *MVS1*, *MVS2*, and *MVS3*.

You should also enter which device number ranges have been or will be defined and paired with Target-ID and LUNs in the attached nodes, for example *500-53F,800-83F*.

The two empty rectangles to the right of each figure should be used to document which CHPIDs, generated as CNC-channels, are attached to which direct ESCON-attached system. From each rectangle (CHPID), draw a line to a port on one of the ESCON-Directors, described under ***B***, or to the ESCON port on a server described under ***E*** below.

The two empty rectangles at the bottom of each figure should be used to document which CHPIDs, generated as CNC channels, are attached to which IBM InfoSpeed Data Gateway. From each rectangle (CHPID), draw a line to a port on one of the ESCON Directors, described under ***B*** below or to one of the ESCON ports on an IBM InfoSpeed Data Gateway, described under ***C*** below.

B On the chart you find two figures representing one ESCON Director each. In the middle of these figures you should enter the ESCON Director number.

In the small boxes in the top row you should enter the ESCON Director port numbers where the ESCON cables from the S/390 systems are connected, see ***A*** above, for example *C1, C7, D3* and so on.

In the small boxes in the bottom row you should enter the ESCON Director port numbers where the ESCON cables from the IBM InfoSpeed Data Gateways or the direct ESCON-attached servers are connected, see ***C*** below, for example *C2, C9, D0* and so on.

C On the chart you find two figures representing one IBM InfoSpeed Data Gateway each. In the middle of these figures you should enter the IBM InfoSpeed Data Gateway serial number, if available. Otherwise select a numbering of your own in order to differentiate between the IBM InfoSpeed Data Gateways.

The two boxes on top of the figures represent the two ESCON ports on the IBM InfoSpeed Data Gateway. From each box (DG1 ESCON port) you should draw a line to a port on one of the ESCON Directors, described under ***B*** above. You may already have drawn connecting lines to the CHPIDs (described under ***A*** above) if no ESCON Director is to be used.

The two boxes at the bottom of each figure represent the two SCSI ports on the IBM InfoSpeed Data Gateway. If an IBM InfoSpeed Multiplexor should be attached you should draw a line from the first box to slot 2 on the attached Multiplexor that will be represented by the figures described under ***D*** below. If a second Multiplexor should be attached you should draw a line from the second box to slot 2 of that Multiplexor. If a UNIX or Windows NT server will be attached directly to the Gateway you should draw a line to the top box in one of the figures described under ***E*** below representing a UNIX or Windows NT server.

D On the chart in Figure 145 on page 170, you find four figures representing one IBM InfoSpeed Multiplexor each. Number the multiplexor in the #-signed box.

The box on top of each figure represents Slot 2 on the IBM InfoSpeed Multiplexor. Slot 2 on the Multiplexor used is to connect the IBM InfoSpeed Data Gateway. From Slot 2 a connection to one of the SCSI ports on an IBM InfoSpeed Data Gateway should already have been drawn as described above under ***C***.

The boxes at the bottom of each row D figure represent Slots 3 through 8 on the IBM InfoSpeed Multiplexor where the UNIX or Windows NT servers are to be attached. Draw a line from each slot to the top box in one of the figures described under ***E*** below, representing a UNIX or Windows NT server.

E On the chart in Figure 145 on page 170, you find 24 figures each representing one UNIX or Windows NT server. In the bottom large rectangle write the name (obtained using the uname command) of the server.

In the box at the top left corner of each E-level figure, document which slot on the server will be used to connect this server to either the IBM InfoSpeed Data Gateway or the IBM InfoSpeed Multiplexor. If you followed the instruction under ***D*** above, you should already have a line drawn between this box and a slot on a Multiplexor or a port on a Gateway. In case of a direct ESCON-attached server you may already have drawn a line directly from the CNC channel on one of the S/390 systems described under ***A*** above. If this server is attached through an ESCON director described under ***B*** above, you need to draw a line to the ESCON director port to be used by this server.

On the lower sixteen of the E-level figures,you find lines drwan already up to the level of the top eight figures, just to make it easier to draw the connections on the chart. In case of a direct ESCON-attached server, document which slot has been used for the ESCON-adapter.

In the box market T-ID, document which SCSI Target-ID will be used for the IBM InfoSpeed Data Gateway from each individual UNIX or Windows NT server.

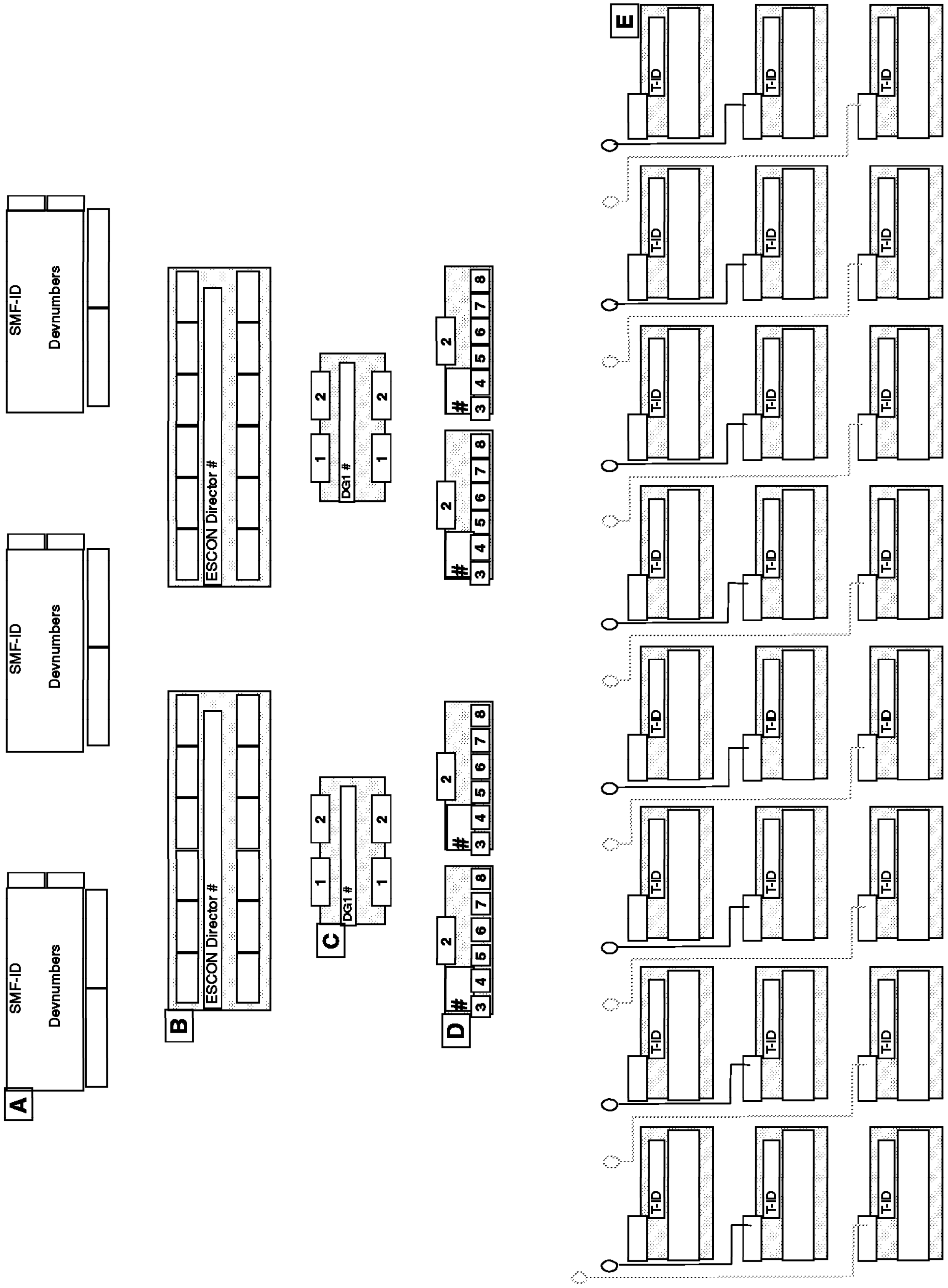


Figure 145. Gateway Configuration Worksheet - Planning Chart

A.1.2 Gateway Configuration Worksheets

The following worksheets should be completed at least 10 working days (ideally more than 15) prior to the arrival of the InfoSpeed Data Gateway. Once these sheets are completed, FAX them to **1.612.797.6804** in the United States. For international customers, the phone number is **xxx1.612.797.6804** -- where xxx is the country′s international calling prefix. Once received, this information will be used to create the configuration that will be needed during the installation of the InfoSpeed Data Gateway.

A.1.2.1 Matrix Information

The matrices in Table 14 on page 172 through Table 18 on page 176, have been copied from the *IBM InfoSpeed 9398 Data Gateway Models DG1 and MX1 Planning Guide*, GA32-0395. They should be completed and FAXed to the above telephone number.

A.1.2.2 Configuration Matrices

The configuration matrices, Table 15 on page 173 through Table 18 on page 176 have areas that are lightly shaded and partly shaded. These areas need not be completed. The lightly shaded areas contai information to be used by the customer — either for reference or for use when FAXing the completed worksheet. The darker shaded areas need no information.

To fax number	1.612.797.6804
Number of pages (Including this page)	

Table 14. Required Customer Information

Customer name:	
Customer address (include city, state and postal code):	
IBM CE contact name:	
IBM CE contact phone number:	
Requested installation date:	
Unit serial number:	
Remote dial in number:	
Completed Configuration Matrices (see Table 15 on page 173, through Table 18 on page 176)	

Table 15. ESCON Port 1 MVS Host Information; Customer name:____

DG1 No. (1)	MVS Path (2)	SMF Host ID (3)	HCD LPAR Num. (4)	HCD CHPID (5)	HCD CHPID Type (6)	HCD CHPID Oper Mode (7)	HCD CUADD (8)	HCD Dev. Model (9)	HCD Device Number (10)	HCD Unit Addr (11)	VM/ESA Virtual Device Address (12)	ESCD Num. (13)	ESCD S/390 Port (14)	ESCD DG1 Port (15)
	Cntrl Path 1				CNC			SCTC						
	Cntrl Path 2				CNC			SCTC						
	Data Path 1				CNC			SCTC						
	Data Path 2				CNC			SCTC						
	Data Path 3				CNC			SCTC						
	Data Path 4				CNC			SCTC						
	Data Path 5				CNC			SCTC						
	Data Path 6				CNC			SCTC						

Table 16. SCSI Information for ESCON Port 1; Customer name:____

SCSI Port PCI-1 or PCI-2 (16)	MX1 Name & Slot # (17)	UNIX/NT Server Name (18)	SP2 I/F Name (19)	H/W Platform (20)	Oper. System (21)	O/S Vers. (22)	PDM FTP Ntwk. Server (23)	PDM FTP Ntwk. Server I/F Name (24)	PDM Path (25)	SCSI Adapter Type (26)	SCSI Target ID (27)	SCSI LUN No. (28)	Vendor ID (29)	Product ID (30)
							Y or N		Cntrl Path 1					
▨	▨	▨	▨	▨	▨	▨	▨	▨	Cntrl Path 2	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 1	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 2	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 3	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 4	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 5	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 6	▨	▨		▨	▨

Table 17. ESCON Port 2 MVS Host Information; Customer name:____

DG1 No.	MVS Path	SMF Host ID	HCD LPAR Num.	HCD CHPID	HCD CHPID Type	HCD CHPID Oper Mode	HCD CUADD	HCD Dev. Model	HCD Device Number	HCD Unit Addr	VM/ESA Virtual Device Address	ESCD Num.	ESCD S/390 Port	ESCD DG1 Port
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
	Cntrl Path 1				CNC			SCTC						
	Cntrl Path 2				CNC			SCTC						
	Data Path 1				CNC			SCTC						
	Data Path 2				CNC			SCTC						
	Data Path 3				CNC			SCTC						
	Data Path 4				CNC			SCTC						
	Data Path 5				CNC			SCTC						
	Data Path 6				CNC			SCTC						

Table 18. SCSI Information for ESCON Port 2; Customer name:____

SCSI Port PCI-1 or PCI-2 (16)	MX1 Name & Slot # (17)	UNIX/NT Server Name (18)	SP2 I/F Name (19)	H/W Platform (20)	Oper. System (21)	O/S Vers. (22)	PDM FTP Ntwk. Server (23)	PDM FTP Ntwk. Server I/F Name (24)	PDM Path (25)	SCSI Adapter Type (26)	SCSI Target ID (27)	SCSI LUN No. (28)	Vendor ID (29)	Product ID (30)
							Y or N		Cntrl Path 1					
▨	▨	▨	▨	▨	▨	▨	▨	▨	Cntrl Path 2	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 1	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 2	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 3	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 4	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 5	▨	▨		▨	▨
▨	▨	▨	▨	▨	▨	▨	▨	▨	Data Path 6	▨	▨		▨	▨

A.1.2.3 Information Matrix Field Definitions

Although HCD is used in the definitions that follow, they apply equally for customers using IOCP to maintain their hardware definitions.

MVS Information Matrix Field Definitions The following definitions apply to the ESCON Ports 1 and 2 MVS Host Information worksheets in the preceding tables.

1. **DG1 No.** — InfoSpeed Data Gateway (DG1) serial number. This will define which DG1 the information is for. After the installation is complete, this will be used to identify this specific DG1.
2. **MVS Path** — Delineates each path that the MVS and open systems server transmit data through. There are two control and six data paths.
3. **SMF Host ID** — Customer′s common name for the host connected to the specified ESCON port.
4. **HCD LPAR Num.** — LPAR number from the customer′s HCD generation. If ESCON Multi-Image Facility (EMIF) is installed and the CHPID used for the DG1 is shared between LPARs, enter 0.
5. **HCD CHPID** — CHPID number connected to the specified ESCON port.
6. **HCD CHPID Type** — Set by IBM to CNC and the customer′s HCD generation ***must*** match.

   Note: If a DG1 is used, then the HCD CHPID definition should be set to CNC. (If this is a S/390 ESCON connection to the server, ***without an InfoSpeed Data Gateway***, then the HCD definition should be CTC.)
7. **HCD CHPID Oper Mode** — This is the operating mode of the ESCON channel and should be set to REC (reconfigurable) or SHR (shared) and ***must*** match the customer′s HCD generation.

   Notes:

   a. If SHR is specified and the Data Gateway has one ESCON port, the UNITADD for each LPAR must be identical.
   b. If SHR is specified and the Data Gateway has two ESCON ports, a second UNITADD range may be specified for the second ESCON port. However, it must map to a separate SCSI target ID and LUN on the open server.
8. **HCD CU Addr** — This is the HCD/IOCP Control Unit Address. The Control Unit Address specified for the Data Gateway and the HCD ***must*** match. If Control Unit Address is not used as part of the CNTLUNIT macro, it defaults to 0.
9. **HCD Dev. Model** — Set to SCTC and the customer′s HCD generation ***must*** match.
10. **HCD Device Number** — This is the device address and corresponds to the ADDRESS parameter from the IODEVICE IOCP macro. There should be up to eight device numbers for each MVS/open system connection — 2-Control and 1-6 Data paths.

    If this data path (column 2) is used by another OS/390 image, specify the SMF-ID of that OS/390 image.
11. **HCD Unit Addr** — This corresponds to the CNTLUNIT UNITADD HCD/IOCP parameter. For example, a unit address of ″00″ can be specified.
12. **VM/ESA Virtual Device Address** — This is the address defined for an MVS guest running under VM/ESA. It may, or may not, be the same as the HCD Device Number, Range value.

13. **ESCD Num.** — ESCON director number, if there is one, otherwise enter NONE. If there are multiple directors, enter the dynamically routed director number.

14. **ESCD S/390 Port** — The ESCON director port number that the host is connected to.

15. **ESCD-DG1-Port** — The ESCON director port number that the DG1 is connected to.

SCSI Information Matrix Field Definitions. The following definitions apply to the SCSI Information for ESCON Port 1/2 worksheets in the preceding tables. This identifies the UNIX/NT server′s view of the Data Gateway.

16. **SCSI Port PCI-1 or PCI-2** — Defines the specific SCSI port on the DG1 to which the ESCON host is transmitting data.

17. **MX1 Name and Slot#** — If applicable, identifies the SCSI multiplexor and the slot number that the server is attached to. Examples are:

    - MX1-1 is the MX1 attached to the DG1′s PCI-port 1.
    - MX1-2 is the MX1 attached to the DG1′s PCI-port 2.
    - ″MX1-1,3″ is the first server attached to the MX1 connected to the DG1′s PCI-port 1, MX1′s slot# 3.
    - ″MX1-1,4″ is the second server attached to the MX1 connected to the DG1′s PCI-port 1, MX1′s slot# 4.
    - ″MX1-2,1″ is the first server attached to the MX1 connected to the DG1′s PCI-port 2, MX1′s slot# 3.

    Note: Slot numbers start with the number 3 and go up to 8 for server attachment. MX1 slot# 2 is always connected to the DG1.

18. **UNIX/NT Server Name** — The customers name for the network server attached to the SCSI port. The name can be obtained by using the uname command. If there is a SCSI multiplexor in use, enter ″MX1.″

19. **SP2 I/F Name** — When an SP2 is used, there needs to be an interface number for the server name associated with it.

20. **H/W Platform** — The architecture of the server hardware. Examples of this would be ″Intel″ and ″RS/6000.″

21. **Oper. System** — The manufacturer and operating system environment (NT or UNIX). An example of this would be ″IBM AIX.″

22. **O/S Vers.** — Version or revision of the operating system defined in column 21. An example of this would be ″4.2.1.″

23. **PDM FTP Ntwk. Server** — This indicates that the server will be configured to support non-PDM nodes downstream from this server. The downstream server is used for TCP/IP or FTP transfers. Enter ″Yes″ or ″No.″

24. **PDM FTP Ntwk. Server I/F Name** — If Column 22 was ″Y,″ then this is used to specify the LAN interface name for the channel-attached node. If applicable, this identifies the name of the LAN interface for the PDM FTP Network Server.

25. **PDM-Path** — Delineates each path that the MVS and open systems transmit data through. There are two control and six data paths. (These are fixed.) Each UNIX tape device specified corresponds to one and only one transfer path between MVS and the open system.

26. **SCSI Adapter Type** — This identifies the type of SCSI interface adapter in the UNIX/NT server. Examples of this are:

- ″F&W DF″ for SCSI 2 Fast and Wide Differential.
- ″F20″ for UltraSCSI which includes SCSI F&W and up to 20 MB transfer rates.

27. **SCSI Target ID** — The SCSI ID that the DG1 will be responding to as the target device for the network server:

    - 0-7 for a Narrow Data bus
    - 0-7 or 8-F for a Wide Data bus

28. **SCSI LUN No.** — SCSI Logical Unit Number, each target ID is capable of only eight LUNs:

    If this data path (column 25) is used by another OS/390 image, specify the SMF-ID of that OS/390 image.

29. **Vendor ID** — Identifies the vendor of the tape device that the DG1 is to emulate. An example of this would be ″Exabyte.″

30. **Product ID** — Identifies the product ID of the tape device that the DG1 is to emulate. An example of this would be ″8500.″

A.1.3 Direct ESCON Attachment Configuration Worksheets

The following matrices should be completed before the ESCON adapter is connected to the OS/390 system and before the definitions are made to the PDM software. The matrices are documentation for your installation only, and need not be faxed to the manufacturing site as for the the Gateway.

A.1.3.1 Configuration Matrices

The configuration matrices, Table 19 on page 180 through Table 21 on page 182, contain lightly shaded and partly shaded areas that need not be completed. The darker shaded areas need no information.

*Table 19. (*_____________) Direct ESCON Adapter UNIX Information; Customer name:____________________*

ESCON Adapter Number (1)	UNIX/NT Server Name (2)	SP2 I/F Name (3)	H/W Platform (4)	Oper. System (5)	O/S Vers. (6)	PDM FTP Ntwk. Server (7)	PDM FTP Ntwk. Server I/F Name (8)
						Y or N	

***** Specify the same server name as in column 2.

*Table 20. (*_____________) Direct ESCON Adapter (Part 1 of __) S/390 Information; Customer name:____________________________*

Local Address (9)	MVS Path (10)	SMF Host ID (11)	HCD LPAR Num. (12)	HCD CHPID (13)	HCD CHPID Type (14)	HCD CHPID Oper Mode (15)	HCD CUADD (16)	HCD Dev. Model (17)	HCD Device Number (18)	HCD Unit Addr (19)	VM/ESA Virtual Device Address (20)	ESCD Num. (21)	ESCD S/390 Port (22)	ESCD Adapter Port (23)
0	Cntrl Path 1				CNC			RS6K						
1	Cntrl Path 2				CNC			RS6K						
2	Data Path 1				CNC			RS6K						
3	Data Path 2				CNC			RS6K						
4	Data Path 3				CNC			RS6K						
5	Data Path 4				CNC			RS6K						
6	Data Path 5				CNC			RS6K						
7	Data Path 6				CNC			RS6K						
8	Data Path 7				CNC			RS6K						
9	Data Path 8				CNC			RS6K						
10	Data Path 9				CNC			RS6K						
11	Data Path 10				CNC			RS6K						
12	Data Path 11				CNC			RS6K						
13	Data Path 12				CNC			RS6K						
14	Data Path 13				CNC			RS6K						
15	Data Path 14				CNC			RS6K						

Table 21. (*____________) Direct ESCON Adapter (Part 2 of 2) S/390 Information; Customer name:____________________________

Local Address (9)	MVS Path (10)	SMF Host ID (11)	HCD LPAR Num. (12)	HCD CHPID (13)	HCD CHPID Type (14)	HCD CHPID Oper Mode (15)	HCD CUADD (16)	HCD Dev. Model (17)	HCD Device Number (18)	HCD Unit Addr (19)	VM/ESA Virtual Device Address (20)	ESCD Num. (21)	ESCD S/390 Port (22)	ESCD Adapter Port (23)
16	Data Path 15				CNC			RS6K						
17	Data Path 16				CNC			RS6K						
18	Data Path 17				CNC			RS6K						
19	Data Path 18				CNC			RS6K						
20	Data Path 19				CNC			RS6K						
21	Data Path 20				CNC			RS6K						
22	Data Path 21				CNC			RS6K						
23	Data Path 22				CNC			RS6K						
24	Data Path 23				CNC			RS6K						
25	Data Path 24				CNC			RS6K						
26	Data Path 25				CNC			RS6K						
27	Data Path 26				CNC			RS6K						
28	Data Path 27				CNC			RS6K						
29	Data Path 28				CNC			RS6K						
30	Data Path 29				CNC			RS6K						
31	Data Path 30				CNC			RS6K						

A.1.3.2 Direct ESCON Adapter Configuration Matrices Field Definitions

Although HCD is used in the definitions that follow, they apply equally for customers using IOCP to maintain their hardware definitions.

***** Specify the same server name as in column 2.

1. **ESCON Adapter Number** — Identifies the ESCON adapter located in the server.
2. **UNIX/NT Server Name** — The customer′s name for the server having the ESCON adapter installed. The name can be obtained by using the uname command.
3. **SP2 I/F Name** — When a SP2 is used, there needs to be an interface number for the server name associated with it.
4. **H/W Platform** — The architecture of the server hardware. Examples of this would be ″Intel″ and ″RS/6000.″
5. **Oper. System** — The manufacturer and operating system environment (NT or UNIX). An example of this would be ″IBM AIX.″
6. **O/S Vers.** — Version or revision of the operating system defined in column 5. An example of this would be ″4.2.1.″
7. **PDM FTP Ntwk. Server** — This indicates that the server is to be configured to support non-PDM nodes downstream from this server. The downstream server is used for TCP/IP or FTP transfers. Enter ″Yes″ or ″No.″
8. **PDM FTP Ntwk. Server I/F Name** — If Column 7 was ″Y,″ then this is used to specify the LAN interface name for the channel-attached node. If applicable, this identifies the name of the LAN interface for the PDM FTP Network Server.
9. **Local Address** — The ESCON adapter can have up to 32 subchannels or local addresses. Each one of them should be paired with an OS/390 device number (Column 18).
10. **MVS Path** — Delineates each path that the MVS and open systems server transmit data through. There are two control and 30 data paths.
11. **SMF Host ID** — Customer′s common name for the host connected to the specified ESCON port.
12. **HCD LPAR Num.** — LPAR number from the customer′s HCD generation. If ESCON Multi-Image Facility (EMIF) is installed and the CHPID used for the ESCON Adapter is shared between LPARs, enter 0.
13. **HCD CHPID** — CHPID number connected to the specified ESCON port.
14. **HCD CHPID Type** — Set by IBM to CNC and the customer′s HCD generation ***must*** match.
15. **HCD CHPID Oper Mode** — This is the operating mode of the ESCON channel and should be set or SHR (shared) and ***must*** match the customer′s HCD generation.
16. **HCD CU Addr** — This is the HCD/IOCP Control Unit Address. The Control Unit Address specified for the ESCON adapter and the HCD ***must*** match. If Control Unit Address is not used as part of the CNTLUNIT macro, it defaults to 0.
17. **HCD Dev. Model** — Set to RS6K and the customer′s HCD generation **must** match.

18. **HCD Device Number** — This is the device address and corresponds to the ADDRESS parameter from the IODEVICE IOCP macro. There should be up to 32 device numbers for each MVS/open system connection — 2-Control and 1-30 Data paths.

    If this local address (Column 9) is used by another OS/390 image, specify the SMF-ID of that OS/390 image.

19. **HCD Unit Addr** — This corresponds to the CNTLUNIT UNITADD HCD/IOCP parameter. For example, a unit address of ″00″ can be specified.

20. **VM/ESA Virtual Device Address** — This is the address defined for an MVS guest running under VM/ESA. It may, or may not, be the same as the HCD Device Number, Range value.

21. **ESCD-Num.** — ESCON director number, if there is one, otherwise enter NONE. If there are multiple directors, enter the dynamically routed director number.

22. **ESCD S/390 Port** — The ESCON director port number that the host is connected to.

23. **ESCD Adapter Port** — The ESCON director port number that the ESCON adapter is connected to.

Appendix B. OS/390 Security Processing in IBM exPress Data Mover

The purpose of this appendix is to describe the new security processing in IBM exPress Data Mover that results for IBM APAR OW34962.

B.1 PDM Security Before APAR OW34962

Currently, PDM requires a higher level of authority than any user for whom it is accessing MVS data sets. Further, if you are using the JES interface, it requires surrogate submit authority. PDM performs a third-party call to RACROUTE to ask the authority of the user at the time request is submitted. You get one of the following codes returned:

418 User is not authorized to access the data set in the manner intended

419 RACF REQUEST=AUTH failed - RACF error

420 Invalid USERID

421 Invalid PASSWORD

419 RACF REQUEST=VERIFY failed - RACF error

Since the first thing that the authorization routine does is check user ID (and optionally password for remote users), if no security package was installed, you got a 422 error.

B.2 PDM Security After APAR OW34962

PDM does not validate userIDs or passwords for MVS users, because they are already logged on or running a batch job that passed userID/password validation. UserIDs and passwords are always validated for remote users.

If no security package is installed, consider user ID and password validated (and not fail with ″422″ error). The 422 error still exists but is used only if the security package returns an internal error. Codes 420 and 421 still exist if the RACROUTE REQUEST=VERIFY tells us the user ID and password are not valid. Error codes 418 and 419 are no longer used.

PDM no longer performs a RACROUTE REQUEST=AUTH at the time the request is submitted.

At the time the request is executed, PDM again issues a RACROUTE REQUEST=VERIFY; however, this time it requests an Accessor Environment Element (ACEE) which represents the security attributes of the user. At any time that it is accessing files (allocating, OPEN, read, write, and so on), it stores this ACEE address in its TCB (Task Control Block), effectively becoming the user for these operations. Whenever it needs to revert to its own security, it clears this pointer. When the transaction is complete, PDM frees the ACEE.

What happens if security fails is that the called system function returns an error code, some of which may be confusing to the end user. So, PDM maps security errors to a new code 246. Code 246 means that the user is not authorized to perform this function. Code 246 is not a reject code but a completion code (since security checking is deferred until the operation is actually attempted).

Using this kind of security has several benefits:

1. PDM requires no special RACF authority other than read access to its datasets and write access to its LOG and QUEUE. (Note that this does not obviate the need for SPDMLINK to be an authorized library, since that need has nothing to do with RACF.)

2. PDM no longer requires surrogate authority for JES, since it IS the user when it submits jobs.

3. PDM operates well in installations without a security package installed.

4. RACF SMF-records are now produced correctly, allowing a RACF administrator to use these records to audit data set access (using the old method, PDM was the user). This might eliminate the need for an installation to define and use PDM′s SMF records.

5. Compliance with an installation′s security definitions is assured (the old method had a few minor holes).

Appendix C. Small Computer Systems Interface (SCSI) Tutorial

Small Computer Systems Interface (SCSI) is fully documented in ANSI standard X3.131-1986. Following the creation of the first SCSI standard, now referred to as SCSI-1, emerging technologies required the expansion of the standard. An enhanced version of the standard was produced in 1994, and is called SCSI-2. Today SCSI-3 standards are under development by the X3T10 committee; however most of the significant changes have already been defined.

The SCSI architecture is device independent, allowing the attachment of any form of device that conforms to the ANSI SCSI standard. With this standard interface, hosts can communicate with intelligent peripheral devices such as fixed disks and tape drives.

The devices are connected by means of a flat multiconductor cable, called the SCSI bus. A twisted-pair cable can also be used, which reduces crosstalk between adjacent signals. A large number of devices of different types can be connected to a host by means of a single SCSI bus cable and a single SCSI attachment feature. This SCSI attachment can be either an external, or integrated adapter and is connected to the host system bus.

A maximum of eight devices (on a narrow bus) or sixteen devices (on a wide bus) can be directly attached to the SCSI bus cable. These devices are of two different types:

- Initiators. The SCSI controller on the RS/6000 is an example of an initiator. It does the bus arbitration and target selection.
- Targets. Disk drives and tape drives are examples of targets. They request transfer to or from the initiator of data, commands, device status, and other items. A target device can support up to eight additional devices, called *logical units*.

At least one of each type of device must be connected to the SCSI bus.

All devices and the attachment features must conform to the ANSI standard, which specifies their mechanical, electrical, and functional requirements, as well as the command set they must support.

SCSI allows several different devices to operate concurrently on a single bus and to share the SCSI bus efficiently. This is done by performing arbitration between any devices that want to use the bus at the same time and managing device disconnect and reconnect.

C.1 Interfaces

Four interfaces are available:

- Intelligent Interface

  SCSI is an intelligent interface that masks:

  – The physical format of the device. The host is concerned only with what type of device is attached for example, disk or tape.

  – The physical configuration of the device. Using logical block addressing, the host requests a file in the form of a logical address that is in the range from zero to the maximum address available on the device.

- The recovery procedures of the device. The device attempts to resolve errors without host intervention.
- The commands used to communicate with the device. The SCSI interface provides generic commands that support many device types; for example, the READ command for a disk device is very similar to the READ command for a tape device. The host requires software that can properly format these commands for each specific device type attached to the SCSI bus. This host software is known as the *Device Driver*.

- Peripheral Interface
  - Up to eight (for narrow bus SCSI) or sixteen (in the case of wide bus SCSI) devices can be attached to a single bus cable. In SCSI-3 an additional cable, called the *Q-Cable*, can be added to allow the attachment of up to 32 devices. These devices can be any combination of initiators (usually host adapters or controllers) or targets (usually peripheral devices)
  - The standard protocol is device independent. The host can attach disk, tape or optical drives and other devices such as printers or scanners to the same bus
  - All SCSI devices are plug compatible.
- Buffered Interface
  - All components can handshake, so that slow or fast devices can properly communicate with slow or fast hosts
  - All transfers consist of real data; data correction, mapping, and caching is done in a buffer.
- Peer to Peer
  - Communication can take the form of:

    Host to host
    Host to peripheral device
    Peripheral device to peripheral device.
  - Each communication (I/O process) is between two devices and each device assumes an operational role. For each communication, one device assumes the role of:

    An Initiator, which is responsible for starting the I/O process, for example *READ*
    A Target, which is responsible for controlling the I/O process.

C.2 Data Transfer

The SCSI bus can transfer data in one of two ways, either in synchronous or asynchronous mode. The mode determines how the data is actually transferred:

- Synchronous — In this mode of operation, each byte of data sent between the initiator and the target must be requested and acknowledged. The series of handshakes entailed makes this the slower of the two modes of data transfer.
- Asynchronous — In this mode of operation, multiple bytes of data can be transferred before acknowledgments are received. Acknowledgments are not received at regular intervals, as in the synchronous mode. At the end of the total transfer operation, the acknowledgments are checked. If the total

number of acknowledgments is as expected, the transfer operation is considered successful.

C.3 Configurations

In a SCSI environment, many different configuration options are possible. The SCSI standard defines three bus configuration options:

- Single initiator and single target
- Single initiator and multiple targets
- Multiple initiators and multiple targets.

The third type of configuration is illustrated in Figure 146. Here, multiple targets are connected to multiple hosts. Although Figure 146 shows all the targets sharing the same bus, you may want to restrict the number of devices to one or two per bus for performance reasons.

In Figure 146, the first two targets could each be a 3570-B01 tape drive, and the last four targets could each be IBM 7204-325 disk drives. The number of logical units is device dependent; a 3570-B01 can have up to two, and a 7204-325 can have only one. The SCSI-2 standard supports up to eight logical devices for each target peripheral controller.

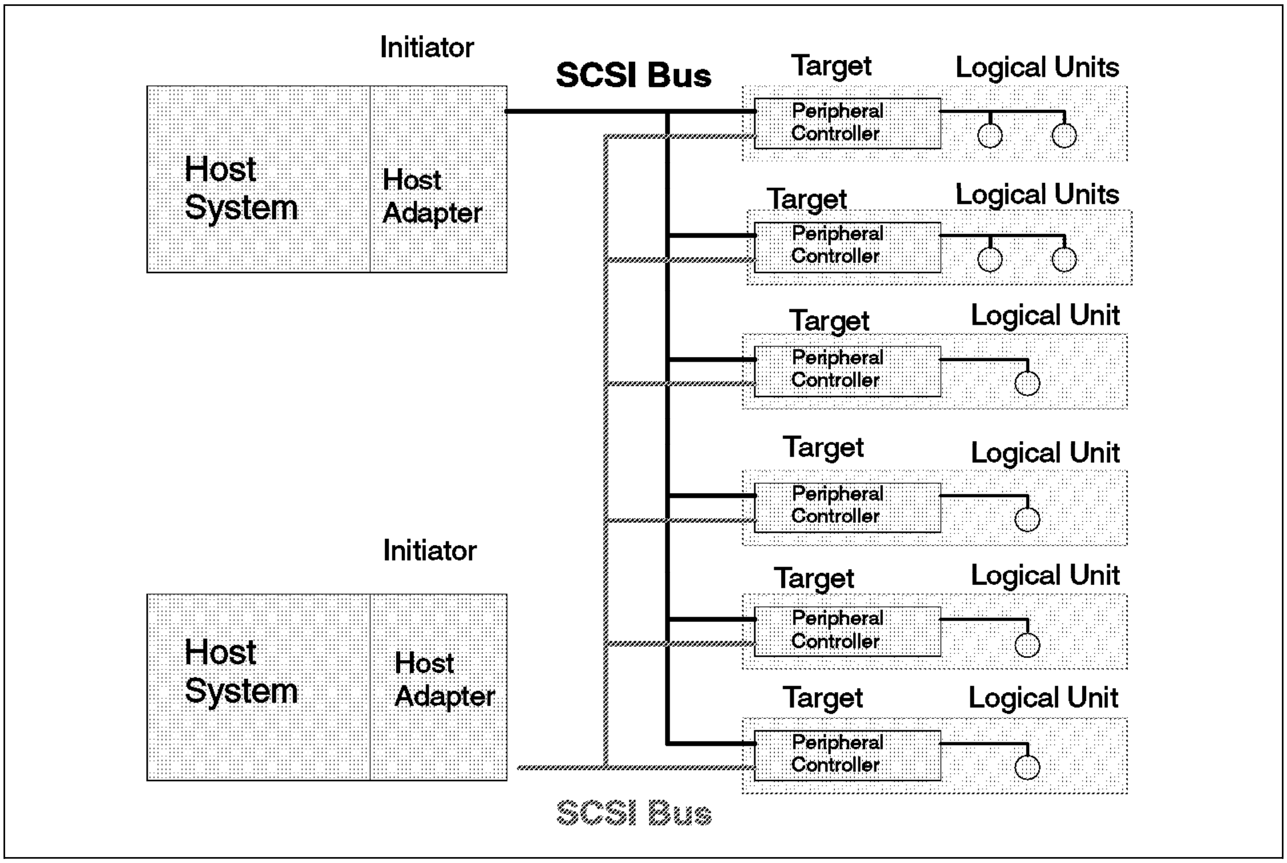


Figure 146. SCSI Multiple Initiators and Multiple Targets

The components shown in Figure 146 are defined as follows:

Host Adapter A device that connects the host bus to the SCSI bus. This is also referred to as the *SCSI attachment* or *SCSI adapter*

Initiator The mode of operation within a device that determines what task needs to be executed and which target is to perform the desired task

Target The mode of operation within a device that controls the entire process, after being successfully selected by an initiator

Logical Unit A physical or virtual peripheral device, addressable through a target or peripheral controller

I/O Process An operation with an input/output device.

C.4 I/O Process

In a SCSI I/O process, three protocol layers may occur:

- Command. This is the transfer operation.
- Data. This is the information to be transferred.
- Status. This updates the initiator that the I/O process has completed.

Figure 147 illustrates the I/O operation protocol layers. These layers are also referred to as *SCSI Bus Phases*. The SCSI bus can be in only one phase at any given time.

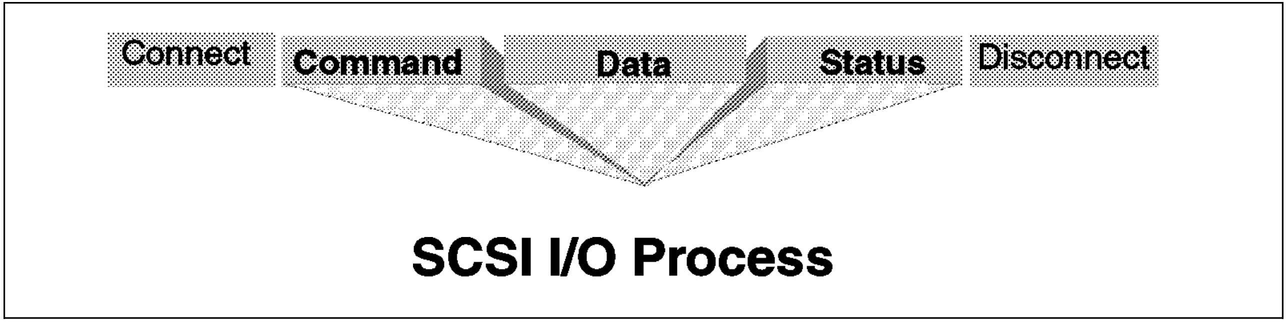


Figure 147. SCSI I/O Operation Sequence

An I/O process (data transfer operation) is initiated after a decision has been made within the host to store or retrieve information to or from an I/O device. Once the host SCSI adapter has the required information, it starts the operation. The total operation includes the initial bus connection and the final bus disconnection. The I/O process comprises the three intervening protocols. Of the three protocols layers defined for an I/O process, at least two — the Command and the Status — must be present. The Data protocol is optional because some SCSI commands have no data to transfer.

C.5 Addressing

In a SCSI configuration, the following components are addressed with SCSI ID numbers:

- Attachment Features — The value of the SCSI ID number is used during arbitration. If more than one device requests the use of the bus at the same time, the one with the highest SCSI ID number is selected. The SCSI attachment feature has a unique SCSI ID number from zero to seven (for narrow SCSI) or zero to fifteen (for wide SCSI) assigned. Typically a SCSI ID of seven (or fifteen) is assigned to the attachment feature, since that is the highest priority ID on the bus.

- Devices — Each device attached to the SCSI bus is recognized as a SCSI device. It is assigned a SCSI ID number that is unique on the bus. This ID is set by means of hardware switches or jumpers located in the device. Most devices have a single SCSI ID; however a device may have more than one SCSI unit. For example, an automated tape library has one SCSI ID for the accessor (or robot) and one for each of the attached tape drives.
- Logical Units — The logical units are not attached directly to the bus, but to the SCSI device. The controllers can be considered SCSI devices and the actual device itself — for example, a disk drive or a tape units can be thought of as the *logical unit*. These logical units can be assigned a SCSI ID number in the range zero to seven. For example, if an IBM 3490-C2A has a SCSI ID of 5, then typically the first tape unit has the SCSI ID set to 0 and the second tape unit has its SCSI ID set to 1.

The SCSI address takes the form “00-**ad**-00-**xy**,” where **ad** is the host adapter number, **x** is the SCSI ID, and **y** is the logical unit number (LUN).

The SCSI attachment feature can support up to seven or fifteen SCSI devices and each device can support up to eight logical control units.

C.6 Tape I/O Commands

The SCSI standard specifies that commands are either optional or mandatory. A device must implement all mandatory commands to conform to the SCSI standard. Below are some typical tape (sequential-access devices) commands:

Mandatory:	READ, WRITE, REWIND
Optional:	LOCATE, WRITE BUFFER, PREVENT MEDIA REMOVAL, ALLOW MEDIA REMOVAL.

Note: For a complete list of tape commands refer to Table 24 on page 197.

Some SCSI tape devices, such as the IBM 3590 Magstar Tape Drive, are also termed *Medium Changer Devices*. Thus in addition to the normal tape commands, these device also support medium changer commands. Below are listed some typical medium changer commands:

Mandatory:	INQUIRY, MOVE MEDIUM, TEST UNIT READY
Optional:	EXCHANGE MEDIUM, POSITION TO ELEMENT, READ ELEMENT STATUS.

Note: For a complete list of medium changer commands, refer to Table 25 on page 198.

C.7 Bus Options

A number of different parameters are used in SCSI terminology:

- Narrow or Wide — describes the width of the data path, either 8-bit (narrow) or 16-bit (wide, but sometimes called half-wide).
- Single ended or Differential — describes how the signals are measured with respect to zero Voltage.
- Normal, Fast, Ultra or Ultra-2 — describes the speed of the bus.
- SCSI-1, SCSI-2 or SCSI-3 — describes the protocols and command sets used.
- Bus Width — The number of bytes that a SCSI bus can transfer in a single I/O operation is termed the *width* of the bus.

  - Narrow Bus — A narrow bus can transfer one byte in a data transfer.
  - Wide Bus — A wide bus can support one or two bytes in a data transfer.

  Both devices and host adapters are termed *Wide or Narrow*, meaning that they transfer two or one byte in a single transfer. Normally, wide and narrow devices can not be mixed on the same SCSI bus.
- Interposer — Typically, not all adapter, device, and cable connectors in a SCSI environment are compatible. To facilitate the connection of the various connector designs, an interposer connector is used. An interposer is a connector that allows the target (device control unit) to attach to the initiator (host adapter).

  It is possible to attach a device to a host adapter where both device and adapter support a different width for data transfer. Thus, for attachment of a 2-byte-wide device to a 1-byte-wide SCSI attachment, a 2-byte-wide interposer is required.
- SCSI Bus Speed
  - Normal Speed SCSI Bus — This bus can transfer data at 4 million transfers per second, which is the SCSI-1 limit.
  - Fast SCSI Bus — This bus can transfer data at 10 million transfers per second, which is the maximum speed of a SCSI-2 bus.
  - Ultra SCSI Bus — This bus can transfer data at 20 million transfers per second.
  - Ultra-2 SCSI Bus — This bus can transfer data at 40 million transfers per second.

  To obtain the effective data rate, multiply the maximum transfer rate by the bus width. For example, for a SCSI-2 Fast/Wide bus,

  10 million transfers per second X 2 bytes per transfer = 20 million bytes per second
- Transceiver Alternatives — There are two types of bus transceiver (driver and receiver) and they are mutually exclusive:

  - Single-Ended — This implementation uses TTL type signal references with +5V being 1 and 0V being 0, with ground being 0V. With a single-ended implementation, the maximum cumulative cable length is 6 m. The distance from the device to the bus can be no greater than 0.1 m, with minimum cable length between the devices of 0.3 m. The single-ended cable option is primarily designed for use within a cabinet.

  - Differential — Pairs of signals are compared with each other. If wire A has a higher voltage than wire B, then that is considered to be a 1. If wire B has the higher voltage, then it is a 0. This technique results in a much higher signal quality than with the single ended approach. Because of the higher signal quality, the maximum cable length is 25 m, with a minimum cable length between the devices of 0.3 m. The distance from the device to the bus cannot exceed 0.3 m. The differential cable option is primarily designed for use outside a cabinet.

  - Cable Options

    A-Cable   This is a 50-pin bus cable, which transfers 1 byte per transfer. It is normally used with SCSI-1

    B-Cable   This is a 68-pin bus cable, which transfers 2 bytes per transfer. The B-Cable is used in addition to the A-Cable. This cable is now obsolete.

P-Cable This is a 68-pin bus cable, which is a Wide 16-bit bus option. This cable can be used as a SCSI1 or SCSI2 bus, but is always required for a SCSI-3 bus.

Q-Cable This is a 68-pin *Full Wide* 32-bit cable. If the Q-Cable is used, then the P-Cable is also required. The P-Cable with the Q-Cable provides a 4 byte-wide bus.

C.8 Other Facilities

As we have seen, more than one initiator can be attached to a single SCSI bus. If one device was a disk and the other a tape, the SCSI attachment could send a command to a tape drive to copy information to the disk device. Once the tape device had read the data it would become an initiator and send a write command to the disk, without further involvement of the SCSI attachment or host.

More than one host can initiate requests on the same bus. Care needs to be exercised when using this arrangement. ADSM for AIX and some target devices do not support this environment.

C.9 Performance

The performance achieved with the SCSI bus, in terms of I/O transactions per second, depends on many factors, including the number of devices attached to the bus, effective use of caching, and the effectiveness of the SCSI attachment feature. An important consideration is the type of operation being performed:

- Disk to tape
- Tape to tape
- Memory to tape.

Data transfer rates in disk-to-tape operations will typically be constrained by the transfer rate of the disk system. Both memory-to-tape and tape-to-tape operations should be able to exploit the potential of the SCSI bus speed more fully.

SCSI implements a bus release feature, *disconnection* which maintains a logical connection between the devices executing a command but releases the bus when one device encounters a delay, such as a full buffer or mechanical seek.

SCSI is a very low overhead architecture, with a protocol time of approximately 17 µs. Actual implementations may vary between 0.5 to 1.5 ms. Very high data transfer rates are possible, up to 40 MB/s, if both 32-bit wide and fast SCSI are implemented. Typical implementations vary between 5 MB/s and 20 MB/s.

C.10 Connecting SCSI Devices

Steps to follow:

1. Plan how you should connect the device in advance. Ensure that disks and tape drives use their own dedicated SCSI attachments and do not share their individual buses with other device types.
2. Determine which SCSI ID (or IDs) the device should have. Make sure you do not use a SCSI ID that is already defined for that bus.

3. Obey the bus cable-length restrictions

4. Draw a map of the bus showing:

   - Every device connected
   - All SCSI IDs
   - Where terminators are connected.

5. You need to install terminators at the extreme ends of the bus. You need to consider the following items when dealing with terminators:

   - The last device connected to the bus (not necessarily the last SCSI ID) must be terminated.

   - Some devices come with internal terminators, which may need to be removed before connection to the SCSI bus

   - Terminators should get their power from the closest device and must have power to operate. Typically, the SCSI attachment feature provides power to one of the terminators. Lack of power to a terminator will cause failures on the SCSI bus.

   - SCSI attachment fuses provide terminator power. These fuses blow if SCSI cables are connected and disconnected without shutting down the host. The result will be faults on the bus and devices not powering-on properly.

6. Check all device SCSI IDs on the bus.

7. Check that the SCSI ID of the attachment feature is set to 7 (or 15), that is the highest priority on the bus.

8. Make sure that you use the correct type of SCSI bus cable or cables.

9. If you need to customize the SCSI bus, be sure to insert the daisy-chain sockets in the correct position.

10. Power down the host before connecting devices.

11. Remove the host SCSI adapter port terminator.

12. Connect the SCSI cable to the host adapter port. Use a twisted-pair cable to reduce crosstalk between adjacent signals.

13. Verify all the connections at this point. For AIX platforms, this can be done by issuing the AIX command DIAG. You must have ROOT authority to execute the DIAG command. The sequence is as follows:

    - Execute the DIAG command
    - Select **Service Aid**
    - Select **SCSI bus**
    - Select the parent adapter, or bus
    - Finally select the SCSI ID of the device you want to verify.

14. Check every SCSI device by selecting its SCSI ID. This sends a Test Unit Ready (TUR) command to that address. If the device is powered on and properly connected, it replies with a valid response, in which case the connection is confirmed. You may receive a valid response to a TUR with a CHECK-CONDITION from the device. This does not necessarily indicate a problem; for example, it could mean that a tape drive does not have a cartridge loaded.

A number of connection problems can occur. Most problems occur at installation, reconfiguration, and the addition of new devices. However, if you follow the above 14-step procedure, many of the usual errors are avoided.

When a problem occurs on a SCSI connection, it pinpoints the exact source, whether hardware or software. A trace of the SCSI bus is typically required for a vendor to identify the nature of the problem.

The Device Driver provides logging to the system error log for a variety of errors. For AIX platforms, view the error log by using the system management interface tool (SMIT) or use the ERRPT command.

C.11 Differences between SCSI-1 and SCSI-2

Table 22 summarizes the difference between SCSI-1 and SCSI-2 commands, advanced functions, and protocol characteristics.

Table 22 (Page 1 of 2). Differences between SCSI-1 and SCSI-2

Device Type	**SCSI-1**	**SCSI-2**
All command types	8/1•	17/4•
CD-ROM commands	15/2	52/8
Communication commands	NONE	20/6
Direct access commands	30/4	45/12
Medium changer commands	NONE	25/5
Optical memory commands	NONE	58/5
Printer commands	16/2	25/7
Processor commands	8/2	15/5
Scanner commands	NONE	26/8
Sequential commands	23/5	33/15
WORM commands	29/3	50/9
Three-Level defect model	NO	YES
Extended inquiry	NO	YES
Extended sense data	OPTIONAL	YES
Log commands	NO	YES
Page mode select/sense	NO	YES
Page mode select/sense command	NO	YES
Read defect command	NO	YES
Arbitration delay specification	2.2 µs	2.4 µs
Arbitration	OPTIONAL	REQUIRED
Single initiator selection	YES	NO
Extended messages	3	4
IDENTIFY message	OPTIONAL	REQUIRED
LUN specified in command	OPTIONAL	NO
Number of messages/mandatory	14/1	22/8

• For the SCSI commands the number following the '/' represents the number of mandatory commands for the SCSI standard.

Table 22 (Page 2 of 2). Differences between SCSI-1 and SCSI-2

Device Type	**SCSI-1**	**SCSI-2**
Number of timing specifications	15	23
Parity	OPTIONAL	REQUIRED

• For the SCSI commands the number following the '/' represents the number of mandatory commands for the SCSI standard.

C.12 Differences between SCSI-2 and SCSI-3

The SCSI-3 standards are still under development by the X3T10 committee. However, most of the significant changes have now been defined. Table 23 is a list of what has significantly changed in SCSI-3.

Table 23. Differences between SCSI-2 and SCSI-3

CHARACTERISTIC	**SCSI-2**	**SCSI-3**
Devices attached to bus	Up to 8	Up to 16, or up to 32 (requires two cables)
Number of logical units	Up to 8	Up to 32
IDENTIFY messages	LIUNTAR	NO - Target routines removed
16 byte command	NO	Group 4 commands
Automatic Contingent Allegiance•	NO	Can replace SCSI-2 Contingent Allegiance
Automatic Sense Capability	NO	Allows Target to automatically return Sense data when Contingent Allegiance condition occurs
LUN in CDB	Ignored if IDENTIFY message was sent	Reserved in all CDB's
INQUIRY command	YES	Enhanced to define new SCSI-3 features
Control byte	YES	Enhanced to handle Automatic Contingent Allegiance
FAST-20	NO	Doubles the rate of the SCSI bus to 20 million transfers per second
CLEAR ACA message	NO	Is used to clear the Automatic Contingent Allegiance condition

• Contingent Allegiance preserves the logical connection to the device following an abnormal condition. The logical connection is preserved until the abnormal condition is cleared. Contingent Allegiance may exist after the following:

- The return of CHECK CONDITION status
- The return of COMMAND TERMINATED status
- An unexpected disconnect, which is optional, not mandatory.

For a summary of the differences between the SCSI-1 and SCSI-2 refer to Table 22 on page 195.

C.13 Command Codes for Sequential-Access Devices

The commands for sequential-access devices, such as tape drives, can be seen in Table 24.

Table 24 (Page 1 of 2). SCSI Commands for Sequential-Access Devices

Command	Operation Code	Type•
CHANGE DEFINITION	40h	O
COMPARE	39h	O
COPY	18h	O
COPY AND VERIFY	3Ah	O
ERASE	19h	M
INQUIRY	12h	M
LOAD UNLOAD	1Bh	O
LOCATE	1Bh	O
LOG SELECT	4Ch	O
LOG SENSE	4Dh	O
MODE SELECT(6)	15h	M
MODE SELECT(10)	55h	O
MODE SENSE(6)	1Ah	M
MODE SENSE(10)	5Ah	O
PREVENT ALLOW MEDIUM REMOVAL	1Eh	O
READ	08h	M
READ BLOCK LIMITS	05h	M
READ BUFFER	3Ch	O
READ POSITION	34h	O
READ REVERSE	0Fh	O
RECEIVE DIAGNOSTIC RESULTS	1Ch	O
RECOVER BUFFERED DATA	24h	O
RELEASE UNIT	17h	M
REQUEST SENSE	03h	M
RESERVE UNIT	16h	M
REWIND	01h	M
SEND DIAGNOSTIC	1Dh	M
SPACE	11h	M

• For sequential access command type:

M Command implementation is mandatory
O Command implementation is optional

The following command codes are vendor specific: 02h, 06h, 07h, 0Ch, 0Dh and 0Eh.

All other command codes for sequential-access devices are reserved for future standardization.

Table 24 (Page 2 of 2). SCSI Commands for Sequential-Access Devices

Command	Operation Code	Type•
TEST UNIT READY	00h	M
VERIFY	13h	O
WRITE	0Ah	M
WRITE BUFFER	38h	O
WRITE FILEMARKS	10h	O

• For sequential access command type:

M Command implementation is mandatory
O Command implementation is optional

The following command codes are vendor specific: 02h, 06h, 07h, 0Ch, 0Dh and 0Eh.

All other command codes for sequential-access devices are reserved for future standardization.

C.14 Command Codes for Medium Changer Devices

The commands for *Medium Changer* devices, such as tape drives, can be seen in Table 25.

Table 25 (Page 1 of 2). SCSI Commands for Medium Changer Devices

Command	Operation Code	Type•
CHANGE DEFINITION	40h	O
EXCHANGE MEDIUM	A6h	O
INITIALIZE ELEMENT STATUS	07h	O
INQUIRY	12h	M
LOG SELECT	4Ch	O
LOG SENSE	4Dh	O
MODE SELECT(6)	15h	O
MODE SELECT(10)	55h	O
MODE SENSE(6)	1Ah	O
MODE SENSE(10)	5Ah	O
MOVE MEDIUM	A5h	M
POSITION TO ELEMENT	2Bh	O
PREVENT ALLOW MEDIUM REMOVAL	1Eh	O
READ BUFFER	3Ch	O
READ ELEMENT STATUS	B8h	O
RECEIVE DIAGNOSTIC RESULTS	1Ch	O
RELEASE	17h	O

• For medium changer command type:

M Command implementation is mandatory

O Command implementation is optional

Command codes C0h through FFh are vendor specific.

All other command codes for medium changer devices are reserved.

Table 25 (Page 2 of 2). SCSI Commands for Medium Changer Devices

Command	**Operation Code**	**Type•**
REQUEST VOLUME ELEMENT ADDRESS	B5h	O
REQUEST SENSE	03h	M
RESERVE	16h	O
REZERO UNIT	01h	O
SEND DIAGNOSTIC	1Dh	M
SEND VOLUME TAG	B6h	O
TEST UNIT READY	00h	M
WRITE BUFFER	3Bh	O

• For medium changer command type:

M Command implementation is mandatory

O Command implementation is optional

Command codes C0h through FFh are vendor specific.

All other command codes for medium changer devices are reserved.

More information on SCSI can be found in *An In-Depth Exploration of Small Computer System Interface*, by David A. Deming, published by *ENDL Publications, Saratoga, California*.

Appendix D. A Short Introduction to DFSMS

The Data Facility Storage Management Subsystem (DFSMS) is an operating environment that automates the management of storage in a S/390 platform. This appendix gives a brief introduction to S/390 data management, DFSMS and the concept of system-managed storage.

D.1 Data Sets

A data set is a collection of logically related data records stored on a volume. A data set can be a source program, for example, a library of macros, or a file of data records used by a processing program.

Data can be stored on a direct-access storage device (DASD) or magnetic tape volume. All types of data sets can be stored on DASD, but only sequential data sets can be stored on magnetic tape. Each block of data on a DASD volume has a distinct location and unique address, making it possible to find any record without extensive searching. Records can be stored and retrieved either directly or sequentially.

Data management is the part of the operating system that organizes, identifies, stores, catalogs, and retrieves all the information that the installation uses. Data management performs these main tasks:

- Sets aside (allocates) space on DASD volumes.
- Automatically retrieves cataloged data sets by name.
- Controls access to data.

D.1.1 Naming Data Sets

Whenever a new data set is allocated, it must be given a unique data set name. Usually, the data set name is given as the *dsname* in JCL.

A data set name can be one name segment or a series of joined name segments. Each name segment represents a level of qualification. For example, the data set name DEPT58.SMITH.DATA3 is composed of three name segments. The first name segment on the left is called the high-level qualifier and the last is the low-level qualifier.

Each name segment (qualifier) is made of one to eight characters, the first of which must be alphabetic (A to Z) or national (# @ $). The remaining seven characters are either alphabetic, numeric (0-9), national, or a hyphen (-). The period (.) separates qualifiers from each other. Including all qualifiers and periods, the length of the data set name must not exceed 44 characters. A maximum of 22 name segments or qualifiers can make up a data set name.

D.1.2 Cataloging Data Set Names

A catalog is a data set that contains information about other data sets. It provides users with the ability to locate a data set by name without knowing where the data set resides. Cataloging data sets, mean that users need to know less about the storage setup.

The catalog describes data set attributes and indicates the volumes on which a data set is located. Data sets can be cataloged through:

- Job control language (JCL) in the disposition (DISP) parameter
- Access method services (ALLOCATE or DEFINE commands)
- Time sharing option (TSO) ALLOCATE command
- Dynamic-allocation (SVC 99) or DYNALLOC macro

For more information on using the JCL DISP parameter, see *MVS JCL Reference* GC28-1757. For a complete explanation of the usage of access method services, the required JCL and examples, see *DFSMS/MVS Access Method Services for ICF* SC26-4906.

D.2 Using DFSMS

DFSMS/MVS uses a catalog and a volume table of contents.

System-managed storage is the IBM automated approach to managing auxiliary storage. It uses software programs to manage data security, placement, migration, backup, recall, recovery, and deletion to ensure that current data is available when needed, and obsolete data is removed from storage.

System-managed storage is tailored to your needs. You define the requirements for performance, security, and availability, along with storage management policies used to automatically manage the direct access, tape, and optical devices used by its OS/390 operating systems.

The combination of system-managed storage and related hardware and software products is called the DFSMS environment.

D.2.1 The DFSMS Environment

To implement the DFSMS environment and to take advantage of all the functions available with OS/390, you need to install a specific set of software products:

Product	**Description**
DFSMSdfp	The functional component of DFSMS/MVS that provides the storage, program, data, and device management functions of OS/390. The Storage Management Subsystem (SMS) component of DFSMSdfp is fundamental in providing these functions.
DFSMSdss	The functional component of DFSMS/MVS that copies and moves data for OS/390.
DFSMShsm	The functional component of DFSMS/MVS that provides the backup, recovery, migration, recall, and space management functions in the DFSMS environment.
DFSMSrmm	The functional component of DFSMS/MVS that provides the management functions for removable media, including tape cartridges and reels, and optical volumes.
DFSORT	Sorts, merges and copies data.
RACF	Controls access to data and other resources in MVS/ESA.

Figure 148 on page 203 shows the relationship among the IBM products that form the DFSMS environment.

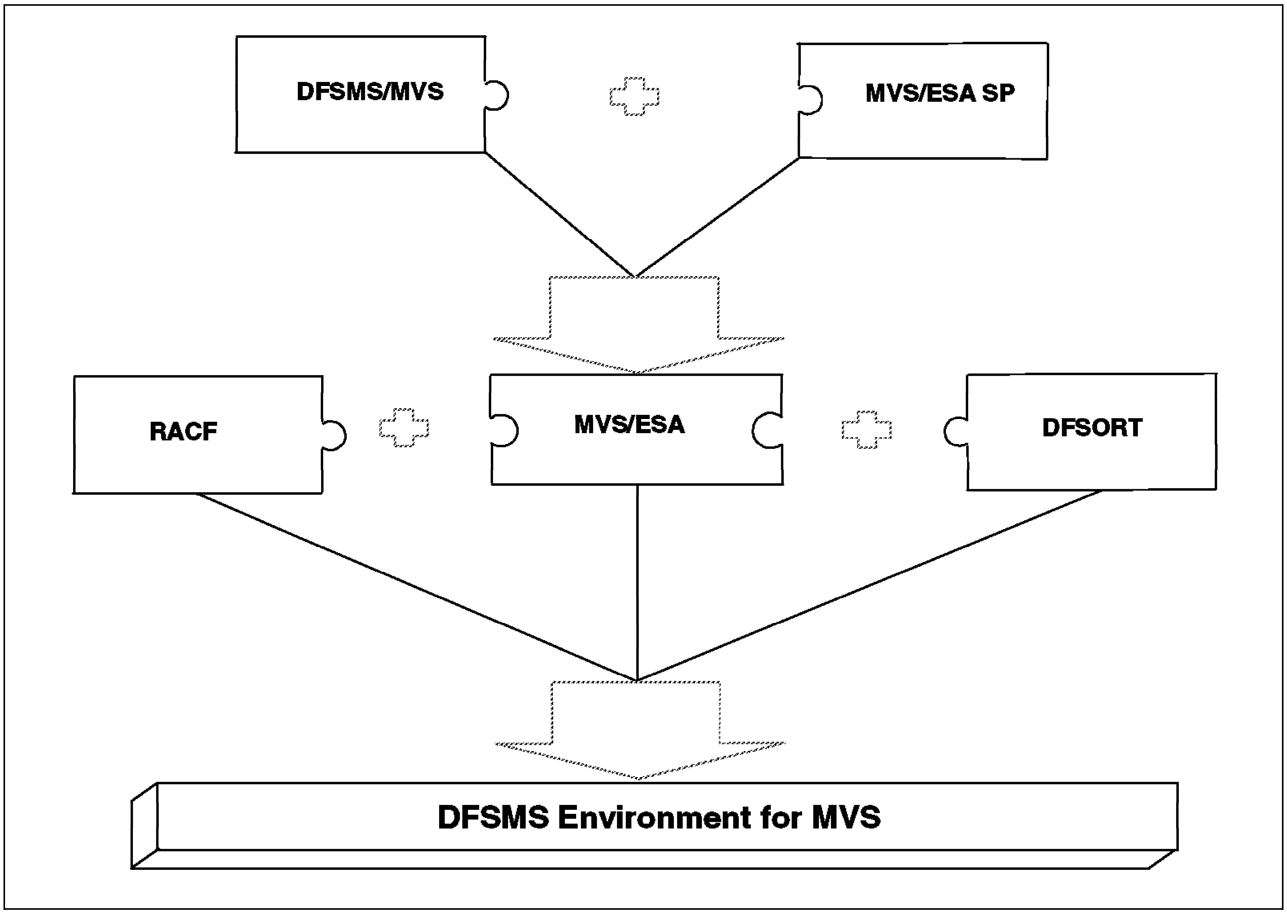


Figure 148. Products that Make Up the DFSMS Environment

D.2.2 Benefits of System-Managed Storage

With SMS, you can define performance goals and data availability requirements, create model data definitions for typical data sets, and automate data backup. SMS can automatically assign, based on installation policy, those services and data definition attributes to data sets when they are created. IBM storage management-related products determine data placement, manage data backup, control space usage, and provide data security.

The goals of system-managed storage are to:

- Improve the use of the storage media; for example, by reducing out-of-space abends and providing a way to set a free-space requirement.
- Reduce the labor involved in storage management by centralizing control, automating tasks, and providing interactive controls for storage administrators.
- Reduce the user′s need to be concerned with the physical details of performance, space, and device management. Users can focus on using data instead of managing data.

The benefits of system-managed storage are:

- **Simplified Data Allocation** — System-managed storage enables users to simplify their data allocations. For example, without SMS, an MVS/ESA user would have to specify the unit and volume on which the system should allocate the data set. The user would also have to calculate the amount of space required for the data set in terms of tracks or cylinders. This means the user would need to know the track size of the device that is to contain the data set.

  With system-managed storage, users can let the system select the specific unit and volume for the allocation. They can also specify size requirements in terms of megabytes or kilobytes. This means the user does not need to know anything about the physical characteristics of the devices in the installation.

- **Improved Allocation Control** — System-managed storage enables you to set a requirement for free space across a set of DASD volumes. You can then provide adequate free space to avoid out-of-space abends. The system automatically places data on a volume containing adequate free space.

  You can also set a threshold for scratch tape volumes in tape libraries, to ensure enough cartridges are available in the tape library for scratch mounts.

- **Improved Input/Output (I/O) Performance Management** — System-managed storage enables you to improve DASD I/O performance across the installation and at the same time reduce the need for manual tuning by defining performance goals for each class of data. You can use cache statistics recorded in system management facilities (SMF) records to help evaluate performance. You can also improve sequential performance by using extended sequential data sets. The DFSMS environment makes the most effective use of the caching abilities of the IBM 3990 Model 3 and Model 6 Storage Controls, as well as other new models.

- **Automated DASD Space Management** — System-managed storage enables you to automatically reclaim space allocated to old and unused data sets or objects. You can define policies that determine how long an unused data set or object will be allowed to reside on primary storage (storage devices used for your active data). You can have the system remove obsolete data by migrating the data to other DASD, tape, or optical volumes, or you can have the system delete the data. You can also release allocated but unused space assigned to new and active data sets.

- **Automated Tape Space Management** — System-managed storage enables you to fully use the capacity of your tape cartridges and to automate tape mounts. Using tape mount management techniques, DFSMShsm can fill tapes to their capacity. With 3490E tape devices, Enhanced Capacity Cartridge System Tape, 36-track recording mode, and the improved data recording capability, you can increase the amount of data that can be written on a single tape cartridge.

  You can also use the IBM 3494 Tape Library Dataserver to automatically mount tape volumes and manage the inventory in an automated tape library. If you do not have an automated tape library data server, you can still take advantage of system-managed tape by using manual tape libraries.

- **Automated-Optical-Space-Management** — System-managed storage enables you to fully use the capacity of your optical cartridges and to automate optical mounts. Using a 3995 Optical Library Dataserver, you can automatically mount optical volumes and manage the inventory in an automated optical library.

- **Improved Data Availability Management** — System-managed storage enables you to use different backup requirements for data residing on the same DASD volume. Thus, you do not have to treat all data on a single volume the same way.

  You can use DFSMShsm to automatically back up CICS/ESA and DATABASE 2 (DB2) databases, partitioned data sets extended (PDSEs), and physical sequential, partitioned, virtual storage access method (VSAM), hierarchical file system (HFS), and direct access data sets. You can also back up other types of data and use concurrent copy to maintain access to critical data sets while they are being backed up. Concurrent Copy, along with Backup-While-Open, has an added advantage that it avoids the invalidation of a backup of a CICS VSAM KSDS due to a control area or control interval split.

  You can also create a logical grouping of data sets, so that the group is backed up at the same time to allow for recovery of the application defined by the group. This is done with the aggregate backup and recovery support (ABARS) provided by DFSMShsm.

- **Simplified Conversion of Data to Different Device Types** — System-managed storage enables you to move data to new volumes without requiring users to update their job control language. Because users in a DFSMS environment do not need to specify the unit and volume that contains their data, it does not matter to them if their data resides on a specific volume or device type. This allows you to easily replace old devices with new ones.

  You can also use system-determined block sizes to automatically reblock those physical sequential and partitioned data sets that can be reblocked.

D.2.3 Managing Data with SMS

In the DFSMS environment, you use SMS classes and groups to set service requirements, performance goals, and data definition models for your installation. You use the Interactive Storage Management Facility (ISMF) to create the appropriate classes and groups, and Automatic Class Selection (ACS) routines to assign them to data according to your installation′s policies.

On systems that do not use DFSMS, storage management consists mostly of manual operations performed on individual data sets, and manual and automated operations performed on volumes. With SMS, you can automate storage management for individual data sets and objects, and for DASD, optical, and tape volumes. You use SMS classes and groups to define the goals and requirements that the system should meet for a data set or object.

Use data class to define model allocation characteristics for data sets; storage class to define performance and availability goals; management class to define backup and retention requirements; and storage group to create logical groupings of volumes to be managed as a unit.

D.2.3.1 Using Data Classes

A *data class* is a collection of allocation and space attributes that you define. It is used when data sets are created. You can simplify data set allocation for your users by defining data classes that contain standard data set allocation attributes. You can use data classes with both system-managed and non-system-managed data sets, but some data class characteristics are only available with system-managed requests.

Data class attributes define space and data characteristics of data sets that are normally specified on JCL DD statements, TSO/E ALLOCATE commands, access method services (IDCAMS) DEFINE commands, dynamic allocation requests, and ISPF/PDF panels. For tape data sets, data class attributes can also specify the type of cartridge and recording method, and if the data is to be compacted. Users then need only specify the appropriate data classes to create standardized data sets.

You can use data class to specify that a sequential data set should be striped. Extended sequential data sets are allocated across DASD volumes to sustain a data rate you specify.

You can also use the data class automatic class selection (ACS) routine to automatically assign data classes to new data sets. For example, data sets with the low-level qualifiers LIST, LISTING, OUTLIST, or LINKLIST are usually utility output data sets with similar allocation requirements, and can all be assigned the same data class.

Even though data class is optional, it is recommended that you assign data classes to system-managed and non-system-managed data. Although the data class is not used after the initial allocation of a data set, the data class name is kept in the catalog entry for system-managed data sets for future reference. The data class name is not saved for non-system-managed data sets, although the allocation attributes in the data class are used to allocate the data set.

If you change a data class definition, the changes affect only new allocations. Existing data sets allocated with the data class are not changed.

D.2.3.2 Using Storage Classes

A *storage class* is a collection of performance goals and availability requirements that you define. It is used to select a device to meet those goals and requirements. Only system-managed data sets and objects can be assigned a storage class. Storage classes free users from having to know about the physical characteristics of storage devices and manually placing their data on appropriate devices.

Some of the availability requirements you can specify with storage classes can be met only by DASD volumes attached through a 3990 Model 3 or Model 6 Storage Control or a similar device. You can specify attributes that require the use of the 3990 Model 3 or Model 6 Storage Control dual copy, DASD fast write, and concurrent copy features. The performance goals can be met through devices attached through storage controls with or without cache, depending on the goals you set.

With storage class, you can assign a data set to dual copy volumes to ensure continuous availability for the data set. With dual copy, two current copies of the data set are kept on separate DASD volumes. If the volume containing the primary copy of the data set is damaged, the companion volume is automatically brought online and the data set continues to be available and current.

You can use the ACCESSIBILITY attribute of the storage class to request that concurrent copy be used when data sets or volumes are backed up.

You can specify an I/O response time objective with storage class. During data allocation, the system attempts to select the closest available volume to the specified performance objective.

For objects, the system uses the performance goals you set in the storage class to place the object on DASD, optical, or tape volumes. The storage class is assigned to an object when it is stored or when the object is transitioned. The ACS routines can override this assignment.

If you change a storage class definition, the changes affect the performance service levels of existing data sets that are assigned that class when the data sets are subsequently opened. However, the definition changes do not affect the location or allocation characteristics of existing data sets.

D.2.3.3 Using Management Classes

A *management class* is a collection of management attributes that you define. It is used to control the retention, migration, backup and release of allocated but unused space for data sets, or to control the retention, backup, and class transition of objects. Management classes let you define management requirements for individual data sets, rather than defining the requirements for entire volumes.

If you do not explicitly assign a management class to a system-managed data set or object, the system uses the default management class. You can define your own default management class when you define your SMS base configuration.

For objects, you can:

- Assign a management class when it is stored, or
- Assign a new management class when the object is transitioned, or
- Change the management class by using the OAM Application Programming Interface (OSREQ CHANGE function).

The ACS routines can override this assignment for objects.

You can use management class attributes to do the following:

- Allow early migration for old generations of a generation data group (GDG) by specifying the maximum number of generations to be kept on primary storage, and what to do with rolled-off generation data sets.
- Delete selected old and unused data sets from DASD volumes.
- Release allocated but unused space from data sets.
- Migrate unused data sets to tape or DASD volumes.
- Specify how often to back up data sets, and whether concurrent copy should be used during backup.
- Specify how many backup versions to keep for data sets.

- Specify how long to save backup versions.
- Specify the number of versions of aggregate backups to keep and how long to retain those versions.
- Establish the expiration date for objects.
- Establish transition criteria for objects.
- Indicate if automatic backup is needed for objects.

By classifying your data according to management requirements, you can define unique management classes to fully automate your data set and object management. For example, you can use management classes to control the migration of CICS user databases, DB2 user databases and archive logs, test systems and their associated data sets, and IMS archive logs. You can specify that DB2 image copies, and IMS image copies and change accumulation logs, be written to primary volumes and then migrated directly to migration level 2 tape volumes.

For objects, you use the class transition attributes to define when an object is eligible for a change in its performance objectives or management characteristics. For example, after a certain number of days you might want to move an object from a high-performance DASD volume to a slower optical volume. You can also use the management class to specify that the object should have a backup copy made when the OAM Storage Management Component (OSMC) is executing.

If you change a management class definition, the changes affect the management requirements of existing data sets and objects that are assigned that class.

D.2.3.4 Using Storage Groups

A *storage group* is a collection of storage volumes and attributes that you define. The collection can be a group of:

- System paging volumes
- DASD volumes
- Tape volumes
- Optical volumes
- Combination of DASD and optical volumes that look alike
- DASD, tape, and optical volumes treated as a single object-storage hierarchy.

Storage groups, along with storage classes, help reduce the requirement for users to understand the physical characteristics of the storage devices that contain their data.

You can direct new data sets to as many as 15 storage groups, although only one storage group is selected for the allocation. The system uses the storage class attributes, volume and storage group SMS status, MVS volume status, and available free space to determine the volume selected for the allocation. In a tape environment, you can also use tape storage groups to direct a new tape data set to an automated or manual tape library.

DFSMShsm uses some of the storage group attributes to determine whether the volumes in the storage group are eligible for automatic space or availability management.

Unlike data, storage, and management classes, users cannot specify a storage group when they allocate a data set, although they can specify a unit and volume. Whether or not you honor their unit and volume request is your decision, but we recommend you discourage users from directly requesting specific devices. It is more effective for your users to specify the logical storage requirements of their data by storage and management class, which you can then verify in the automatic class selection routines.

D.2.3.5 Using Automatic Class Selection Routines

You use ACS routines to assign class and storage group definitions to data sets, database data, and objects. You write ACS routines using the ACS language, which is a high-level programming language. Once written, you use the ACS translator to translate the routines to object form so they can be stored in the SMS configuration.

The ACS language contains a number of read-only variables, which you can use to analyze new data allocations. For example, you can use the read-only variable &DSN to make class and group assignments based on data set or object collection name, or &LLQ to make assignments based on the low-level qualifier of the data set or object collection name. You cannot alter the value of read-only variables.

You use the four read-write variables to assign the class or storage group you determine for the data set or object, based on the routine you are writing. For example, you use the &STORCLAS variable to assign a storage class to a data set or object.

For a detailed description of the ACS language and its variables, see *DFSMS/MVS DFSMSdfp Storage Administration Reference*, SC26-4920.

For each SMS configuration, you can write as many as four routines: one each for data class, storage class, management class, and storage group. Use ISMF to create, translate, validate and test the routines.

Figure 149 on page 210 shows the order in which ACS routines are processed. Data can become system-managed if the storage class routine assigns a storage class to the data, or if it allows a user-specified storage class to be assigned to the data. If this routine does not assign a storage class to the data, the data cannot reside on a system-managed volume.

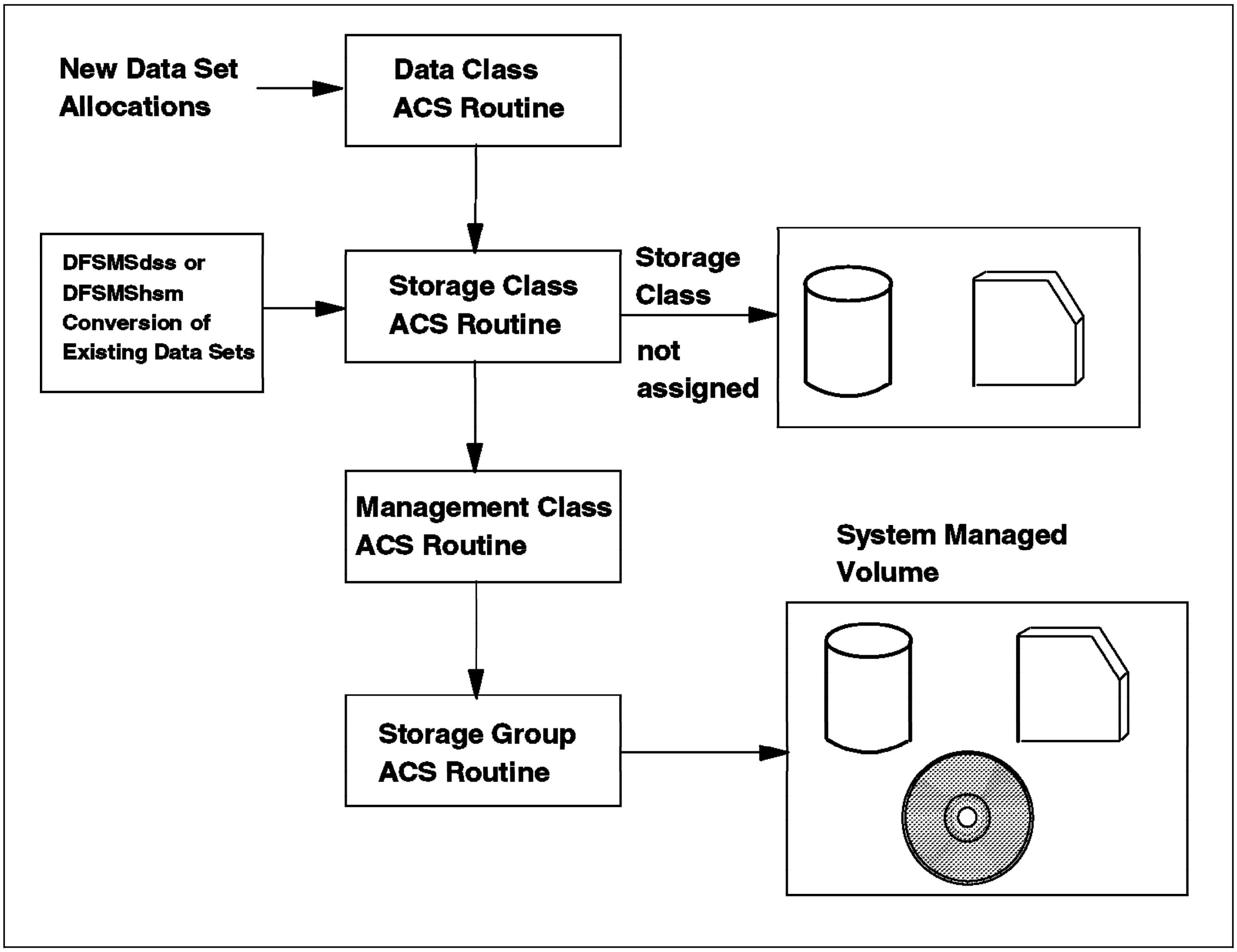

New Data Set Allocations
Data Class ACS Routine
DFSMSdss or DFSMShsm Conversion of Existing Data Sets
Storage Class ACS Routine
Storage Class not assigned
Management Class ACS Routine
System Managed Volume
Storage Group ACS Routine

Figure 149. Diagram Showing Processing of ACS Routines

Because data allocations, whether dynamic or through JCL, are processed through ACS routines, you can enforce installation standards for data allocation on system-managed and non-system-managed volumes. ACS routines also enable you to override user specifications for data, storage, and management class, and requests for specific storage volumes.

You can use the ACS routines to determine the SMS classes for data sets created by the Distributed FileManager/MVS. If a remote user does not specify a storage class, and if the ACS routines decide that the data set should not be system-managed, the Distributed FileManager/MVS terminates the creation process immediately and returns an error reply message to the source. Therefore, when you construct your ACS routines, consider the potential data set creation requests of remote users.

Appendix E. UNIX basics

This appendix describes the very basic commands an OS/390 system engineer has to know to configure a UNIX PDM channel-attached node or PDM network-attached node. We assume for now that you are using a UNIX system that was set up by someone else. Usually, you will just have to log on, go to the proper directory, and follow the installation steps as described in Chapter 5, “Install and Configure IBM InfoSpeed” on page 31. But in the event of a fingercheck, or any kind of unexpected accident, it is always good to know a little more.

E.1 Login to the system

When the system detects that you have established a new connection to it (either by rebooting the machine itself or by dialing into it), it issues a *login* prompt. To respond, type your user name the press Enter. The system then prompts you for your password; type it, then press Enter. At this point, your interaction with UNIX is controlled by a program called a *shell* To issue a command to the shell, type the command and follow it by pressing Enter. The shell will execute the command and send its output to your terminal.

To terminate your session and log out, type exit, logout or Ctrl-D. The command to be used depends on the UNIX system you are using, so try them all.

You will find hereunder some of the most useful commands you will need to install and configure the IBM InfoSpeed Data Gateway, the IBM InfoSpeed Multiplexor and the IBM exPress Data Mover

E.2 Go to the Proper Directory

Use these commands:

- *pwd* tells you in which directory you are. If you issue this command just after you logged on, it should answer you */home/user_name*.
- *df* shows all the file systems that are available on your machine. Typically, you should have */home, /usr, /tmp, /var*, which are the basic file systems of a UNIX system. Other file systems may have been added by the UNIX administrator, such as */cdrom, /oracle, /sap* or others, according to what the machine is used for and which products are installed.

  You should consider these file systems as base directories or as PC disk drives (*C:, D: ...*) for our purpose.
- *cd* allows you to move inside file systems and directories, just as in DOS or Windows:
  - *cd /usr* takes you to the /usr file system
  - *cd /usr/lpp* takes you the the directory *lpp* in the */usr* file system. If you already are in the */usr* directory, just type *cd lpp*, with no */*.
  - *cd ..* takes you back to your previous directory
- *ls* lists the content of the directory you are in. It lists the whole content, making no distinction between files, subdirectories or links. If you want to

know which is a directory and which is not, you need to issue the *ls -al* command.

- *ls -al* describes the content of the directory. The letter in the first column indicates the type of file you are considering : d is a directory, - is a file, l is a link, s is a xxxxx. This letter is followed by nine characters that describe rights. These can be r (for read), w (for write), x (for execute), or - (for no right). They are given for the user of the file, for its owner, and for the owner′s group. For example, if you have something like:

  – *-r--r-xrwx 1 root system 955363 Dec22 data.file*

  This line means that the file *data.file* was created on December 22, has a size of 955363 bytes, is owned by *root*, who is a member of the *system* group. The user only has the right to read, *root* has the right to read, write (that is modify) and execute the file (since it is recognized as being an executable file), but other members of the system group can only read and execute, but not modify it.

- *id* tells you who you are (user name) and which group you belong to.

- *ps -ef* or *ps -ag* tells you which programs are currently running on your system. If you don′t want to see all processes, but only those that concern PDM (dmeexec, dmeroute ...), you have to pipe this command into a grep command that selects only the processes you are interested in:

  – *ps -ef | grep dme*

E.3 Launch Interactive Management Menus

AIX and HP/UX both provide very useful menus to install and configure software and hardware pieces on your machines. The menus are called respectively SMIT (Software Maintenance Interface Tool) and SAM (System Axxx Maintenance). Most of the panels shown in this menu are self-explanatory. Please refer to each specific paragraph of Chapter 5, “Install and Configure IBM InfoSpeed” on page 31 to learn how to launch and use these for your use.

Here are some examples of fast access commands to reach a specific SMIT menu :

- *smitty users* to add or modify users
- *smitty groups* to add or modify a group
- *smitty esca* to add or modify an ESCON subchannel

E.4 Use the Editor

During the installation of IBM exPress Data Mover, the software detects which text editor you have available on your machine. In the graphical desktop environment, which is commonly installed on AIX machines, there is a dtpad text editor, which is easy to understand and use. But in most cases, you will probably have to enter into the editor *vi*.

A summary of commands to create, modify, delete, locate, move and display text using the screen editor vi is in E.4.1, “How to Use vi” on page 213. To give you a flavor of the vi editor here are some typical vi commands:

- *vi file_name*: opens the requested file, and puts you in a browsing mode. You can move into the file by using the arrows, the backspace key, and the space bar, but you cannot write anything.
- *Esc*-key and *i* are complementary commands once you opened a file with vi. Command *i* places you in insert mode, so that you can write in the file. The *Esc*-key places you back in the browsing mode.
- *:q, :q!, or :wq!*. When you are in Browsing Mode, you can close the file by typing *:q* or *:q!* . You just quit the file, without saving any of the modifications you have made. If you want to save the modifications, you need to write and save, by issuing a *:wq!* command. *You must be in browsing mode (Escape key) for issuing these commands*.

E.4.1 How to Use vi

This section is a summary of commands to create, modify, delete, locate, move and display text using the screen editor vi.

E.4.1.1 Invoking vi

To enter the editor, enter *vi number file* where the optional parameter *number* is where the cursor will be placed as a starting position in the *file*. To view contents of a file enter *vi -R file* or *view file*. To edit multiple files, enter *vi file1 file2 file3*.

Editor *vi* can run with different options. To display the setting of the vi. options, enter *:set all*. To set an option, type *:set option value*. Some common options are:

- *autowrite* - sets autowrite feature on
- *ignorecase* - ignores cases in searches
- *number* - numbers each line
- *showmode* - flags input mode
- *wrapmargin n* - sets margin to *n*

E.4.1.2 vi Commands

Input Mode Commands

a	Appends text immediately after the cursor
i	Inserts text immediately after the cursor
o	Appends text after current line
O	Appends text before current line
Esc	Exits input mode and goes to browse mode
Backspace	Deletes current input character
^*w*	Deletes current input word
@	Deletes current input line

Replacing Text

*r*c	Replaces current character with specified character ″c″
S	Replaces current line
R	Replaces with ″string″ from cursor until the *Esc*-key is entered
s	Deletes current character then appends text until the *Esc*-key is entered
~	Switches between lower and upper case
cw	Replaces current word (from cursor to next space)

Undoing Changes

u	Undoes most recent delete or change
U	Undoes all changes on the current line
:e!	Undoes all changes since last write

Moving and Copying

*:r*file	Reads in ″file″
n*Y*	Yanks ″n″ lines
″a*y*	Yanks into buffer ″a″
p	Puts yanked text immediately after cursor or below current line
P	Puts yanked text immediately before cursor or above current line

Deleting Text in Command Mode

x	Deletes current character
n*dd*	Deletes ″n″ lines (default is current)
D	Deletes from cursor to end of line
dw	Deletes current word (from cursor to next space)
cw	Replaces current word (from cursor to next space)

Cursor Positioning Commands

Window Positioning

H	Home (Moves cursor to the top left of the screen)
M	Moves cursor to the middle of the screen
L	Moves cursor to the bottom left of the screen

Character Positioning

h	Moves cursor one character to the left
Backspace	Moves cursor one character to the left
l	Moves cursor one character to the right
Space	Moves cursor one character to the right
*f*x	Moves cursor right to character ″x″
*F*x	Moves cursor left to character ″x″
*t*x	Moves cursor right just before character ″x″
*T*x	Moves cursor left just after character ″x″

Word Positioning

w	Moves cursor to next word
e	Moves cursor to end of word
b	Moves cursor to beginning of previous word

Line Positioning

j	Moves cursor down one line
k	Moves cursor up one line
^	Moves cursor to beginning of line
$	Moves cursor to end of line
+	Moves cursor to beginning of next line
-	Moves cursor to beginning of previous line

Sentence and Paragraph Positioning

(	Moves cursor to beginning of sentence
)	Moves cursor to next sentence
{	Moves cursor to beginning of paragraph
}	Moves cursor to beginning of next paragraph

Screen Positioning

^*F*	Moves cursor forward one full screen
^*D*	Moves cursor forward one half screen
^*B*	Moves cursor back one full screen
^*U*	Moves cursor back one half screen
^*G*	Moves cursor to last half screen (end of file)

Searching

*/*string	Searches for next appearance of ″string″
*?*string	Searches backward for first appearance of ″string″
n	Repeat last search
N	Repeat last search in opposite direction

Line Editor Commands

:!	Escapes to shell temporarily
*:*n	Moves cursor to line ″n″ of buffer
:$	Moves cursor to last line in buffer
*:r*file	Inserts ″file″ after current line
*:s/*a*/*b*/*	Replaces ″a″ with ″b″
*:*a,b*w*file	Writes lines ″a″ through ″b″ to ″file″

Saving

:w	Writes the file
:n	Makes next file the current edit file

Special Commands

.	Repeats last command
^*l*	Clears and redraws current window
J	Joins current line with following one

Exiting

:wq	Writes file and quits
ZZ	Writes file and quits
:q	Quits if changes were written
:q!	Quits unconditionally

E.4.1.3 Use of Macros With vi

To use macros in command mode, enter *:map name commands* where *name* is a one-character or special-character key (function key) which is mapped to a series of specified commands. For example *:map Q :q!* will cause an immediate exit from *vi* when character *Q* is entered in command mode.

To use macros in input mode, enter *:map! name string* where *name* is a one-character or special-character key (function key) which is mapped to the string. For example *:map! x ed* will insert the characters *ed* when the character *x* is entered in input mode.

To unmap macros in command mode, type *:unmap name*, and to unmap macros in input mode, type *:unmap! name*.

Appendix F. Very Short FTP Introduction

This chapter describes the TCP/IP File Transfer Protocol and its main commands. Don′t forget that IBM exPress Data Mover uses FTP-like transfer protocols, which contain most FTP typical commands, plus some others specific to PDM. When you launch a DMEFTP interface, it does not use TCP/IP but a channel-to-channel (CTC) communication protocol. Only the commands and the way you use the FTP interface look the same.

The *TCP/IP FTP* program enables you to transmit files to and from a remote computer that understands FTP, and to operate on files and directories there. FTP is a command interpreter. It processes commands that you provide either interactively or from a file and translates them into instructions to the server that handles the lower-level FTP commands. To open a session, issue the following command:

- *ftp host_name* or
- *ftp IP-address*

For example:

- *ftp aixserv* or
- *ftp aixserv.almaden.ibm.com* or
- *ftp 9.101.212.13*

If the command is successful, it should issue you an *ftp>* prompt. On many systems you must use the *user* command to supply a user ID and password. Now you can start to transfer data, by using the following commands :

- *cd* establishes a default directory on the remote node to simplify your later FTP commands
- *lcd* establishes a default directory on the local node to simplify your later FTP commands. When you start to transfer data, FTP can transfer it into or out of the directory specified by the last *lcd* command.
- *binary* or *ascii* allows to specify whether you want a binary or an ASCII file transfer. In TCP/IP FTP, ascii is the default but binary is the default in IBM exPress Data Mover.
- *get file_name* starts the transfer of the requested file into the directory specified by the last *lcd* command. You can issue a *mget ** to transfer all the files of the remote directory you are in. FTP asks you to confirm every file transfer, unless you first issued a *prompt* command:
- *put file_name* starts the transfer of the local requested file into the remote directory specified by the last *cd* command
- *help* prompts a very rich help menu
- *quit* makes you quit the FTP interpreter

The TCP/IP FTP always transmits the files in serial mode. You cannot request parallel transfer, as you can in IBM exPress Data Mover. In IBM exPress Data Mover, you can also request checkpoints to be taken during file transfers. No such function is available in TCP/IP FTP.

Appendix G. Special Notices

This publication is intended to help storage administrators, systems programmers, and system administrators to implement IBM InfoSpeed Solutions. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM 9398-DG1 InfoSpeed Data Gateway, IBM 9398-MX1 Multiplexor for the IBM 9398-DG1 InfoSpeed Data Gatway, IBM 5697-D61 IBM InfoSpeed exPress Data Mover V1 for OS/390, and IBM 5648-PDM IBM InfoSpeed exPress Data Mover for UNIX and Windows NT. See the PUBLICATIONS section of the IBM Programming Announcement for IBM 9398-DG1 InfoSpeed Data Gateway, IBM 9398-MX1 Multiplexor for the IBM 9398-DG1 InfoSpeed Data Gatway, IBM 5697-D61 IBM InfoSpeed exPress Data Mover V1 for OS/390, and IBM 5648-PDM IBM InfoSpeed exPress Data Mover for UNIX and Windows NT about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR®
AIX®
AS/400®
BatchPipes®
CICS/ESA®
DATABASE 2
DB2®
DFSMS
DFSMS/MVS®
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
ESCON®
IBM®
IMS
IMS/ESA®
Magstar
Multiprise
MVS/ESA
OpenEdition®
OS/390
Parallel Sysplex
PROFS®
RACF
RAMAC
RISC System/6000®
RS/6000
S/390®
Seascape
SNAP/SHOT®
SP
SP2
System/390
S/390 Parallel Enterprise Server
VM/ESA®
3090

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Appendix H. Related Publications

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

H.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 225.

- *Data Sharing: Cross Platform Extension (XPE) Implementation Guide*, SG24-5256
- *Data Sharing: Using the OS/390 Network File System*, SG24-5262

H.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

H.3 Other Publications

These publications are also relevant as further information sources:

- *IBM InfoSpeed 9398 Data Gateway, Models DG1 and MX1, Planning Guide*, GA32-0395
- *InfoSpeed exPress Data Mover, Version 1 Release 1, Installation and Customization Guide*, GA32-0389
- *InfoSpeed exPress Data Mover, Version 1 Release 1, User′s Guide and Reference*, GA32-0388

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download or order hardcopy/CD-ROMs redbooks from the redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders via e-mail including information from the redbook order form to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755	
Canada (toll free)	1-800-IBM-4YOU	
Outside North America	(long distance charges apply)	
(+45) 4810-1320 - Danish	(+45) 4810-1220 - French	(+45) 4810-1270 - Norwegian
(+45) 4810-1420 - Dutch	(+45) 4810-1020 - German	(+45) 4810-1120 - Spanish
(+45) 4810-1540 - English	(+45) 4810-1620 - Italian	(+45) 4810-1170 - Swedish
(+45) 4810-1670 - Finnish		

This information was current at the time of publication, but is continually subject to change. The latest information for customers may be found at http://www.redbooks.ibm.com/ and for IBM employees at http://w3.itso.ibm.com/.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may also view redbook, residency and workshop announcements at http://inews.ibm.com/.

IBM Redbook Fax Order Form

Fax your redbook orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

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

List of Abbreviations

ACS	automatic class selection
ADSM	ADSTAR Distributed Storage Manager
APA	all points addressable
APAR	a program analyses report
CTC	channel to cahnnel
DASD	direct access storage device
DCB	data control block
ERP	enterprise resource planning
ESCON	Enterprise Storage Connection
FTP	file transfer program
GDG	generation data group
HCD	host configuration definition
HFS	hierachical file system
IBM	International Business Machines Corporation
IOCP	input output control program
IPL	initial program load
ISPF	interactive system productivity facility
IT	information technology
ITSO	International Technical Support Organization
JCL	job control language
JES	Job Entry Subsystem
MVS	Multiple Virtual Storage
OPC	Operation Planning and Control
PDM	exPress Data Mover
PDS	partioned data set
PDSE	partioned data set extended
PROFS	Professional Office System
RACF	Resource Access Control Facility
RMM	Removable Media Manager
RRDS	relative record data set
S/390	System/390
SAF	System Authorization Facility
SCSI	small computer storage interface
SMF	system management facility
SMIT	system maintenance interface
SMP/E	system maintenance program extended
SSI	subsystem interface
TCB	task control block
TCP/IP	Transmission Control Protocol/Internet Protocol
TSO	time sharing option
VSAM	virtual storage access method
VTS	virtual tape server

Index

Numerics

3490 cartridge
  physical 144
  virtual 144
5645-001 16
5648-PDM 13
5655-068 16
5655-069 16
5697-D61 13
5799-CSI 43

A

ABARS 205
abbreviations 227
abend 61
ABEND S013 127, 128
abnormal termination 61
ACCEPT 46
ACCEPT CHECK 46
account information file 163
accounting information 163
ACEE 185
acronyms 227
ACS 127, 205
adapter 36
  Adaptec AHA-2940U 17
  Adaptec AHA-2944W 17
  EISA SE A2679A 17
  HP 28696A 17
  HSC FWD A4107A-0D1 17
  IBM FC2412 17
  IBM FC2416 17
  P/N 21490030 Rev B (H) 32
  P/N 21490054 Rev A 31
  Sun X1062A 17
ADDRESS 177, 184
address range 65
addressing
  SCSI 190
  SCSI ID 190
ADSM 10, 157, 158
AIX 4.2 17
AIX only 13, 19, 27, 28
ALLOCATE 202
allocation errors 127
alternate path 21
APAR OW33336 44
APAR OW34962 185
APF 47
application conversion 150
application-to-application 4, 7
APPLY 46
APPLY CHECK 46
archiving 8, 10
ASCII 6, 16, 50, 54, 106, 125, 127, 131, 151, 152
ASCII NOTRIM 131
asynchronous 123, 164
ATYPE 54
authorization 49
authorized library 61
Automated DASD Space Management 204
Automated Optical Space Management 204
Automated Tape Space Management 204

B

BAAN 157
Backup-While-Open 205
backup/recovery 9, 10
backup/restore 8, 10
bandwidth 20, 34, 52, 139
basic configuration 49
bat file 27
batch 27, 51, 106, 109
BatchPipes 133
bibliography 223
BINARY 54, 131
BLKSIZE 54, 127, 128
BLOCKS 128
BSAM 27, 106, 124, 157
bulk file transfer 3
bus option
  fast SCSI bus 192
  narrow bus 191
  SCSI 191
  wide bus 192
business continuation 9
business intelligence 1, 6, 7, 135
business recovery 8

C

cancel 105, 117, 118, 119, 120
cancel request 114
cancel transaction 49
case insensitive 55, 107
case sensitive 107
Catalog Search Interface 43, 132
cataloged 129
CDEF 51
CDEF Statement 53, 55, 62, 65
centralized management 10
cfgmgr 71
channel addresses 62, 63
channel-attached host 15
channel-attached nodes 19, 21, 23, 25, 27, 28, 29, 30, 44, 48, 49, 53, 55, 99, 101, 103

checkpoint 119
Checkpoint/restart 131
checkpointing 127, 129
checksum 52
CHKPT 107
CHPID 44, 167
CICS VSAM KSDS 205
CICS/ESA 205
CLEAR 128
CLIST 50, 108, 110, 111
CLPA 48
cluster configuration 5
CNC 44, 177
CNC channels 167
CNTLUNIT 44, 177, 184
COMM 52, 53
command
  account 26
  ascii 26
  binary 26
  CAN 120
  cd 26
  close 26
  CTA 120
  display 60
  echo 26
  get 26
  getx 26, 27, 123
  lcd 26
  ldelete 26
  lsite 107, 108
  modify 55, 61
  mt 64, 71, 78, 83
  nosync 26
  notify 26
  OFF 120
  ON 120
  open 26
  parallel 26
  put 26
  putx 26, 27, 123
  query 26
  serial 26
  SETSSI 60
  SETSSI ADD 60
  STA 121
  START 61
  STOP 61
  STP 121
  sync 26
  uname 52, 168
command language 105
command library 106, 109, 110
communication driver 53
communication path 53
communication paths 21, 28, 30, 48, 65, 75, 80, 86
communication protocol 53
communication tasks 51, 52, 53, 55, 62, 119, 121
completion code 117
components 13, 14
Concurrent Copy 205
concurrent transfer 61
concurrent transfers 29, 30, 44, 49, 52, 61
concurrent writing 130
configuration
  define 36
  diagram 36
  download 41
  file 73, 79, 85, 94, 101
  loading 41
  ordering 35
  remote update 41
  SCSI 189
  table 29
connections
  gateway 34
  multiplexor 34
control paths 14, 28, 29, 65
Copy between UNIX or Windows NT 150
CTDRV 53
cumulative-service 46
CYLINDERS 128

D

daemon 28, 103, 104
DALEXPDL 131
DALSSNM 124, 125
DALSSPRM 124, 125
data class 107, 108, 142, 163, 205
Data in Virtual 48
data integrity 5
data management 10
data mining 6, 7, 9
data paths 14, 28, 29, 30, 52, 65, 75, 86, 139, 183
data sharing
  heterogeneous 1
  homogeneous 1
  network-based 1
  storage subsystem-based 1
data transfer
  SCSI 188
data transfer devices 44
data warehousing 9
data-copy sharing 1, 2, 3, 8, 9, 10, 13, 135
database 3, 8
  backup 159
  extraction program 6, 133
  load 8, 150
  load program 6, 133
  unload 150
DATACLAS 27, 128
DATATYPE 127, 155
DATATYPE=A 151
DATATYPE=N 152

DB2 205
DB2 image copies 208
DCB 127, 151, 153
dd 148, 159
DDDEFs 46, 47
Debug Level 94, 102
default allocation file 163
Default allocation for FTP 49
DEFINE 202
DELE 52
Dell Dimension 17
device numbers 28, 29, 44, 45, 52, 53, 55, 120, 167
DFSMS 107, 163, 201
DFSMS/MVS 1.3 157
DFSMS/MVS Version 1 Release 4 43
DFSMSdfp 202
DFSMSdss 202
DFSMShsm 202, 205, 209
DFSMSrmm 202
DFSORT 202
DG1 No 177
differential cable 34
DIRBLKS 54, 128, 130
directly-attached nodes 28
directory
/etc/rc2.d/S99pdm 87
directory listings 43
disaster recovery 10
Disclaimer
measured performance 136
disk 25, 128
distance 35
Distributed FileManager/MVS 210
distribution libraries 45, 46, 47
distribution tape 43
DMEACCHK 46
DMEACCKM 46
DMEACCUM 46
DMEACPDM 46
DMEALLOC 47
DMEAPCHK 46
DMEAPCKM 46
DMEAPCUM 46
DMEAPPDM 46
DMECLIB 109
DMECTCTC 53
DMEDDDEF 46, 47
DMEExec 96, 100
DMEFTP 105, 106
DMEFTP service 104
DMEINITQ 48
DMEIOCTC 52
DMEIOCTS 52
DMERCCUM 46
DMERCPDM 46
DMERJPDM 46
DMERoute 96, 100
DMERXLAT 50
dmeserv 30
DMESETUP 46, 47, 50
DMESISPF 50, 105, 111
DMESMF 50
DMESYS 48, 61, 62
DMETABLE 48, 51, 52
DMETBASE 49
DMETFTP 49
DMETNET 49
DMETONE 49
DMETTWO 49, 55
download
microcode 31
downloading 120
downstream nodes 24, 30, 53, 97, 101, 132, 178, 183
DSCB 128, 130
DSNAME 124, 127
DYNALLOC 202
dynamic allocation 26, 106, 124, 125, 126, 127, 202
dynamic LPA 48

E

e-business 1, 135
EBCDIC 6, 16, 50, 106, 125, 127, 151, 152
EMIF 19, 21, 183
ENTRIES= 48
environment file 75, 76, 77, 83, 94
ERP 157
error code 117
Errors 121
ESCD Adapter Port 184
ESCD DG1 Port 178
ESCD Num 178, 184
ESCD S/390 Port 178, 184
ESCON 7, 9, 13, 14, 15, 16, 17, 19, 20, 21, 22, 25, 28, 29, 63, 65
adapter 19, 69, 168
Adapter Number 183
attachments 29
cables 167
channels 35, 65
connectors 21
director 19, 21, 22, 35, 36, 167, 178
ESCON 14
port 14, 31
ports 34, 167, 168
ESCON 1 31
ESCON 2 31
ESCON-attached 27, 28, 29, 44, 167
Event Manager 27, 55, 105, 111, 114, 117, 118, 119, 120, 121
Exabyte EXB-8500 87
Exabyte2 device driver 87
executable 27, 30, 133, 154, 155, 157
EXPDT 128, 131

F

FAX
  Outside USA 171
  United States 171
Feature
  2710 33
  2711 33
  2712 33, 34
  2756 17
  4001 34
  4002 34
  4003 34
  4010 34
  5000 34
  5001 34
  7703 34
  7704 34
  7705 34
features 19, 33
fiber definitions 28, 70
file
  /etc/inetd.conf 100, 103, 104
  /etc/inittab 76, 100
  /etc/rc.local 81
  /etc/services 100, 103
  /etc/services/ 104
  /kernel/drv/st.conf 82
  /opt/dme/config/dme_init.d 87
  /reconfigure 82
  etc/init.d/dme 87
  install_pdm 72
  size 76
FILETYPE 127
FILETYPE=X 154
free standing 34
FTP 13, 16, 24, 49, 52, 53
FTP commands 217
FTP protocol 52
FTP TCP/IP interface 28
full-volume backup 157
function 117

G

GDG 27, 112, 130
GDG base entry 130
generally available 141
GETX 150

H

H/W Platform 178, 183
Hardware Configuration Definition 44
HCD 43, 44, 60, 63, 65, 177
HCD CHPID 177, 183
HCD CHPID Oper Mode 177, 183
HCD CHPID Type 177, 183
HCD CU Addr 177, 183
HCD Dev Model 177, 183
HCD Device Number 184
HCD LPAR Num 177, 183
HCD Unit Addr 177, 184
HDEF 51
HDEF Statement 52, 55, 65, 99
HDETABLE 43, 48, 50, 60, 62, 63, 106, 119, 120
heterogeneous 2, 4
HFS 157, 205
hierarchical storage management 10
high-level qualifier 43
homogeneous 2, 4
HOST 127
HP-UX
  Version 10.2 17, 76
  Version 10.20 17, 76
HP/9000-D220/D270-D271 17
HPDM110 46, 47

I

I/O driver 52
I/O process
  SCSI 190
IATPLBnn 48
IBM 3088 44
IBM 3494 9, 157, 204
IBM 3900 9
IBM 3990 206
IBM 3995 Optical Library Dataserver 204
IBM 7133 8
IBM 9398-DG1 13, 136
IBM 9398-MX1 13, 135
IBM standard labels 131
IDCAMS 127, 150
IEAAPFxx 47
IEASYSxx 45
IEBGENER 127, 131, 148, 150, 151
IECIOSxx 45
IEFDB476 127
IEFSSNxx 45, 60
implementation 34
Improved Allocation Control 204
Improved Data Availability Management 205
Improved I/O Performance Management 204
IMS image copies 208
inetd 100, 103, 104
information network 10
input/output devices 43
install
  hardware 41
  software 41
interactive 109
interface
  SCSI 187
internal reader 131
internet 24

intranet 24
IOCP 43, 44, 60, 177
IODEVICE 44, 177
IODRV 52
IPL 45
IRXISPRM 51
IRXPARMS 51
IRXTSPRM 51
ISMF 205
ISPF 27, 50, 60, 105, 111, 115
ISPF client 50

J

JES interface 185
JES job submission 49
JES mode 156
JES option 150
JES2 48
JES3 48
JOBLIB 51

L

LAN interface 178, 183
large file support 76
LE/370 43
limitation 22
LINKLIB 47
LNKLST 48, 51
Local Address 183
local MVS calls 123
locking 5
LOG dataset 186
log out 211
login 211
Lotus Notes 157
LPALSTxx 48
LPARs 21, 183
LRECL 54, 127, 128

M

maintenance level 49
management class 10, 205
MAXT 51, 61
MAXT Statement 52, 55
megabytes transferred 117, 120
message
  IEA101I 48
MGMTCLAS 27, 128
microcode
  download 31
Microsoft Windows NT Version 4.0 87
MIGRAT 131
migrate/recall 8
migrated data sets 131
migration level 2 208
Missing Interrupt Handler 43, 45, 60
MKDIR 146
MODEL 128
modem 32, 33
modem connection 31
modem switch 32, 33, 34
monitoring 27, 105
multihost 21
multiple interfaces 29, 139
multiple requests 166
multiple transfers 25
multiserver 21
MVS File Transfer Client 26
MVS Path 177, 183
MVS PDM transfer task 29, 30
MVS UNIX 156
MVS/ESA SP Version 5 Release 2.2 16, 43

N

named pipe 158
network 10
network computing 6, 7
network-attached nodes 23, 24, 25, 28, 29, 30, 49, 97, 99, 101
NIS services table 103
Node File Transfer Client 26
non-PDM nodes 24, 25, 30, 49, 178, 183
non-SMS 127
nonprimary PDM subsystem 45
nonsynchronous 165
NOSYNC 106
NUMUNIT 128

O

O/S Vers 178, 183
OAM 207
OFF 128
offline 120
ON 128
on-line transaction processing 8
On/Off switch 31
one-to-one communication 20, 21
online 120
OpenEdition 156
Oper System 178, 183
OS 52, 53
OS/390 image 167
OS/390 Version 1 Release 3 16, 43
OS/390-centric 25
OSMC 208

P

PACKTB_HEADER 51
PACKTB_LOCAL_TOTAL 51
PACKTB_LOCAL_USED 51

PACKTB_NEXT_LOCAL 51
parallel 28, 106, 108, 130, 145, 165
parallel mode 166
Parallel Sysplex 2, 5
parallel transfer 135, 166
parallelism 34
partition 2
PASS 127, 155
PATH 124, 127, 154, 157
path number 120
PCI 1 31
PCI 2 31
PDM channel-attached nodes 19, 20, 21, 22, 23, 24, 25, 30, 104
PDM Distribution Tape 43, 47
PDM File Transfer Client 25, 26
PDM FTP Network server 27, 28, 30
PDM FTP Ntwk Server 178, 183
PDM FTP Ntwk Server I/F Name 178, 183
PDM FTP server 103, 104
PDM ISPF Client 26, 27
PDM MVS communication access server 25, 29
PDM MVS File Transfer Client 26, 105
PDM MVS FTP Client 43
PDM MVS ISPF client 52, 105
PDM MVS remote access server 25, 29
PDM MVS server 27
PDM MVS subsystem 25, 28, 29
PDM MVS supervisor 25
PDM Network FTP Server 30
PDM Node File Transfer Client 26, 30
PDM node router 19, 24, 25, 29, 30
PDM Node Subsystem 26
PDM Node Transfer Client 30
PDM Node transfer task 25
PDM Path 178
PDM REXX 27
PDM REXX Interface 26, 51
PDM router 22, 23
PDM SubSystem Interface 26, 106
PDM TCP/IP FTP server 24
PDM UNIX or Windows NT software 25, 28
PDM UNIX or Windows NT transfer task 29, 30
PDS 27, 54, 112, 125, 129, 166
PDSE 27, 54, 125, 129, 205
peer-to-peer communication 19
performance 6, 14, 19, 20, 23, 136
  SCSI 193
phone connection 31
phone line 41
pipe 30
  standard 133
piping 6, 7, 11, 106, 133, 154, 155, 157
Piping data between applications 150
Piping data between UNIX servers 150
pkgadd 81, 82
planning 167
port 58 103, 104
port 59 103, 104
power connections 31
power supply 34
power switch 41
prefix 47
prefixing 132
PRIMARY 54, 128
primary PDM subsystem 45
printing
  high volume 9
PRIV 51
PRIV Statement 52, 55
privileged user 49, 51, 55, 119, 120
Product ID 179
PROFILE 125, 155
profiles 125
Program Directory 43, 46
PROGxx 47
PUTX 150

Q

QSAM 27, 106, 124, 157
QSIZE 53
QUERY DETAIL REQID 156
QUERY RESULT ALL 146
queue 30, 43, 48, 60
queue data set 48, 186

R

RACF 16, 43, 185, 186, 202
RACF REQUEST=AUTH 185
RACF REQUEST=VERIFY 185
RACF SMF records 186
rack mounted 34
RACROUTE 49, 185
RAS 53
raw partition 159
reason code 117, 127
REC 177
RECEIVE 46
Receive Buffers 69
RECFM 54, 127, 128
redundant configuration 21
region size 52, 61
REJECT 46
Remote support 33
REN 52
request ID 62
request number 117
restore 138
RETPD 128
RETR 52
return code 127
REXX interface 27, 60, 123
REXX script 123

RMM 157
RRDS 131
RS/6000 19
  PS580/E30 17
RS6K 44, 183
RS6K-2 44
RVA 8

S

SAF interface 43, 49
SAM 212
SAP 157
script 27, 109
SCSI 9, 13, 14, 15, 16, 20, 22, 25, 28, 29, 63, 77, 80, 81, 82, 86
  adapter 17, 77, 82
  Adapter Type 178
  addressing 190
  attached 44, 52
  attachment 3, 29
  bus option 191
  bus transceiver 192
  cables 34
  command 190
  command codes for medium changer devices 198
  command codes for sequential-access devices 197
  configurations 189
  connecting devices 193
  data 190
  data transfer 188
  device 190
  differences between SCSI-1 and SCSI-2 195
  differences between SCSI-2 and SCSI-3 196
  differential 192
  ESCON converter 14
  fast SCSI bus 192
  host adapter 189
  I/O process 190
  Information Matrix Field Definitions 178
  initiator 189
  interface 187
  interposer 192
  logical unit 190, 191
  LUN No 179
  narrow bus 191
  performance 193
  port 14, 15, 31, 34
  single-ended 192
  slot 34
  status 190
  tape I/O commands 191
  target 190
  Target ID 179
  terminators 63
  tutorial 187
  wide bus 192
SCSI-2 28
SCTC 44, 177
SDEF 51
SDEF statement 55
SDMECLST 50
SDMELINK 43, 47, 60, 61
SDMELOAD 48
SDMEMACS 50
SDMEPROC 48
SDMESAMP 48, 49, 50, 55, 123
SDMEXLTB 50
Seascape architecture 9
SECONDARY 54, 128
security checking 49
security product 43
SERIAL 106, 108
Serial 1 31, 32, 33
Serial 2 31, 32, 33
session 24
SET BDY 47
SET IOS=xx 45
SET PROG=xx 47
SETSSI 45, 60
Shared application data 150
shell 211
SHR 177, 183
Simplified Conversion of Data to Different Device Types 205
Simplified Data Allocation 204
Simultaneous unload/load of databases 150
sizing 167
slot numbers 178
slots
  slot 1 32
  slot 2 32
  slot 3 32
  slot 4 32
  slot 5 32
  slot 6 32
  slot 7 32
  slot 8 32
SmartBatch 133
SMF 51, 60
SMF accounting 49, 50, 55
SMF Host ID 177, 183
SMF records 50, 55, 186
SMF Statement 52, 55
SMFID 74, 80, 86, 95, 99, 101, 167, 179
smit 65, 69, 75, 212
smitty 212
  esca 65
  groups 65
  update_all 65
  users 65
SMP/E 43, 46, 47
SMP/E zones 47
SMS 27, 127

SnapShot 8
SOD 157
Solaris
  Version 2.4 17, 81
  Version 2.5 17, 81
  Version 2.5.1 17, 81
SOURCE 106, 109, 110, 111
SP2 I/F Name 178, 183
space management 10
specifications 34
SSI 8, 26, 27, 106, 124, 133
SSN= 45
standard pipe 133
start 121
start time 117
started task 29
state 120
Statement of Direction 10, 157
status 115, 116, 117
STEPLIB 51
stop 121
stop time 117
STOR 52
storage class 205
storage group 205
storage media 25
storage sharing 2
STORCLAS 27, 128
subchannel
  addresses 75, 83
subchannels 29, 30, 65, 68, 70, 183
SUBHOST 52, 53
submit 156, 158
submit authority 185
submit job 150, 160
SUBSYS 45, 124, 125, 127
SUBSYS= 60
subsystem calls 123
SUM 52
Sun Sparc5 17
surrogate authority 186
SURROGATE class 49
surrogate submit authority 185
SVC 99 126, 202
swcopy 76
swinstall 76
SYNC 106, 108, 146
synchronous 123, 146
SYS1.CMDPROC 50
SYS1.LPALIB 51
SYS1.PARMLIB 43, 45, 47, 48, 60, 167
SYS1.PROCLIB 48
SYS1.SAMPLIB 51
SYSMOD 47
SYSPROC 50
system-determined block sizes 205
system-managed storage 201
System/390 16
SYSTSPRM 51

T

tape 25
  library 9, 10
  management 7, 9
  tape library 157
tape I/O commands
  SCSI 191
tape management systems 131
TAPE<n> 128
tar 132, 147, 158, 159
Target ID/LUN 28, 29, 63, 77, 82, 87, 95, 167, 169
TCB 185
TCP/IP 7, 9, 10, 11, 13, 16, 24, 30, 49, 53
TCP/IP File Transfer Protocol 217
TCP/IP FTP client 30
TCP/IP FTP interface 30
TCP/IP FTP language 105
TCP/IP FTP server 30, 104
TCP/IP socket 30
telecommunications protocol
  DS1 137
  DS3 137
  T-1 137
  T-3 137
terminators 34
third-party call 49, 185
throughput 25, 120
TRACKS 128
trailing blanks 152
transactions 25, 29, 30
transfer paths 29, 51, 52, 62, 80, 86, 114, 119, 120
transfer tasks 28, 29, 30, 49, 62, 119, 121
Transmit Buffers 69
true data sharing 2
truncating 152
TSO 51, 106, 108, 115, 149, 202
TSOREXX1 51
TSOREXX2 51
TSOREXX3 51
TYPE 53
TYPE=CNC 44
TYPE=RS6K 44
TYPE=RS6K-2 44
TYPE=SCTC 44

U

unallocate 120
uncataloged 129
UNIT 52, 53, 54, 55, 128
UNIT=RS6K 44
UNIT=SCTC 44
UNITADD 177
universal data access 7

UNIX commands
- cd 211
- df 211
- id 212
- ls 211
- ls -al 212
- ps -ef 212
- pwd 211

UNIX/NT Server Name 178, 183
unnamed pipe 158
uploading 120
upper case 112, 131
USER 127, 155, 156

V

Vendor ID 179
VERIFY 129
vi commands 212, 213, 214, 215
vi editor 212
vi macros 215
Virtual Tape Server 157
virtual tapes 157
VM/ESA guest 177, 184
VM/ESA Virtual Device Address 177, 184
VOLUME 54, 128
volume backup 159
VS COBOL II Run Time Library 43
VSAM 131, 205
VTS 7, 9, 10

W

Wildcards 116, 129
Windows NT
- Version 4.0 17
- Version 4.00.1381 17

working directory 132

X

XLATTAB 50

ITSO Redbook Evaluation

Data Sharing: Implementing InfoSpeed Copy- Sharing Solutions Among Heterogeneous Systems
SG24-4120-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and Fax it to: USA International Access Code + 1 914 432 8264 or:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer   __Business Partner   __Solution Developer   __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:
____

____

____

____

What other redbooks would you like to see published?
____

____

____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**
____

____

____

____

____

Data Sharing: Implementing InfoSpeed Copy- Sharing Solutions Among Heterogeneous Systems

SG24-4120-00

SG24-4120-00
Printed in the U.S.A.