

IBM Ultrium Device Drivers



Installation and User's Guide

IBM Ultrium Device Drivers



Installation and User's Guide

Note!

Before using this information and the product that it supports, be sure to read the general information under Appendix D, "Notices," on page 301.

Fifteenth Edition (May 2005)

| This edition of the *IBM Ultrium Device Drivers Installation and User's Guide*, GA32-0430-11 obsoletes and replaces all earlier versions. Changes or additions are indicated by a vertical line in the left margin.

© **Copyright International Business Machines Corporation 2000 - 2005. All rights reserved.**

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
-------------------	----

Tables	xi
------------------	----

Preface	xiii
-------------------	------

Related Information	xiii
IBM TotalStorage Ultrium External Tape Drive 3580	xiii
IBM TotalStorage Ultrium Tape Autoloader 3581	xiii
IBM TotalStorage Ultrium Tape 2U Autoloader 3581	xiii
IBM TotalStorage Ultrium Tape Library 3582	xiii
IBM TotalStorage Ultrium Scalable Tape Library 3583	xiii
IBM TotalStorage UltraScalable Tape Library 3584	xiii
Other Publications	xiv
Additional Information	xiv
How to Send Your Comments	xv

Summary of Changes	xvii
------------------------------	------

Fifteenth Edition (May 2005)	xvii
Fourteenth Edition (December 2004)	xvii
Thirteenth Edition (May 2004)	xvii
Twelfth Edition (April 2004)	xvii
Eleventh Edition (October 2003)	xvii
Tenth Edition (June 2003)	xvii
Ninth Edition (April 2003)	xvii
Eight Edition (January 2003)	xvii
Fifth Edition (March 2002)	xvii
Fourth Edition (September 2001)	xviii
Third Edition (July 2001)	xviii
Second Edition (March 2001)	xviii

Part 1. Introduction to IBM Ultrium Device Drivers 1

Chapter 1. Ultrium Device Drivers. 3

Purpose	3
Platform Support	3
Introduction	4
IBM TotalStorage Ultrium External Tape Drive 3580	6
IBM TotalStorage Ultrium Tape Autoloader 3581	6
IBM TotalStorage Ultrium Tape 2U Autoloader 3581	6
IBM TotalStorage Ultrium Tape Library 3582.	6
IBM TotalStorage Ultrium Scalable Tape Library 3583	7
IBM TotalStorage UltraScalable Tape Library 3584	7
StorageSmart by IBM Ultrium Products	7

Part 2. AIX Tape and Medium Changer Device Driver 9

Chapter 2. Introduction and Product Requirements 11

Purpose	11
Data Flow	11
Product Requirements	11
Software Requirements	11
Library Control Path and Data Path Failover Support.	11
Hardware Requirements	12

Chapter 3. Tape Drive, Media, and Device Driver Parameters. 15

Configuration Parameters.	15
Autoloading	15
Block Size	16
Compression	16
Logging	16
Maximum Size of the Log File	16
Record Space Mode	16
Rewind Immediate	17
Trailer Labels.	17
Media Parameter	17

Chapter 4. Installation and Configuration Instructions 19

Installation Procedure	19
Preinstallation Considerations	19
Installation Procedure	20
Configuring Tape and Medium Changer Devices	20
Deconfiguring Tape Devices	21
Deconfiguring Medium Changer Devices	21
Uninstallation Procedure	21

Chapter 5. Special Files 23

Special Files for Tape Devices	23
Special Files for Medium Changer Devices	25

Chapter 6. Alternate Pathing Support for Library Control Path Failover 27

Configuring and Unconfiguring Alternate Pathing Support	28
Primary and Alternate Paths.	28
Querying Primary and Alternate Path Configuration	28
Configuring and Unconfiguring Primary and Alternate Devices	29

Chapter 7. Alternate Pathing Support for Tape Drives 31

Automatic Failover	31
------------------------------	----

Dynamic Load Balancing	32
Installing Feature Code 1681 Data Path Failover License Key	32
Configuring and Unconfiguring Alternate Pathing Support	33
Primary and Alternate Paths.	33
Querying Primary and Alternate Path Configuration	34
Configuring and Unconfiguring Primary and Alternate Devices	34

Chapter 8. Using the Dump Support . . . 35

Chapter 9. Tape Utility Program (tapeutil) 37

Interactive Interface	37
Command-Line Interface	37
General Subcommands	38
Medium Changer Subcommands	40
Tape Subcommands	41
Service Aid Subcommands	45
Block Size and SCSI Transfer Size	45
Configuration Parameters.	45
Reserve and Release Commands	46
Tape Drive Service Aids	46
Volume ID for Logging	46

Chapter 10. Tape Drive Service Aids . . . 47

Details of Tape Service Aids	47
Force Microcode Dump	47
Read Dump	48
Microcode Load	48
Error Log Analysis	49
Reset Drive	49

Chapter 11. Performance Considerations 51

Data Path	51
Common AIX Utilities.	51
Before Calling Support	52

Chapter 12. Device and Volume Information Logging 53

Log File	53
Tape Log Utility.	53

Chapter 13. Problem Determination . . . 55

Error Logging	55
Error Log Templates	55
Error Labels	55
Detail Data	55
Automatic Dump Facility for the IBM 3580 Ultrium Tape Drive	57
Trace Facility	57
ATRC Utility	57

Part 3. Compaq Tru64 Tape and Medium Changer Device Driver. . . . 59

Chapter 14. Introduction and Product Requirements. 61

Purpose	61
Product Requirements	61
Hardware Requirements	61
Software Requirements	61
Setting Up the Environment.	61

Chapter 15. RAS Utility Program For Compaq Tru64 System (IBMrasutil) . . . 63

Installation Procedure from the Tape Device Drivers and SMI-S Agent CD	63
Installation Procedure from the Device Driver FTP Site	63
Uninstalling	64
Update Procedure	64
Query Procedure	64
Verify Procedure.	64
Interactive Mode	64
Command-Line Mode	64
Command-Line Options	65

Part 4. HP-UX Tape and Medium Changer Device Driver 67

Chapter 16. Introduction and Product Requirements. 69

Purpose	69
Product Requirements	69
ATDD Implementation	69
Hardware Requirements	69
Software Requirements	72
Data Flow	73
Software Interface to the Device Driver	74

Chapter 17. Install, Uninstall, and Configure the DLKM Version of ATDD . . . 75

Create the Drive Configuration File (Optional).	76
Determine the Drive Hardware Path for IBM 3580 Ultrium Tape Drive, 3581 Tape Autoloader with SCSI Attachment	76
Determine the Drive Hardware Paths for IBM Ultrium Tape Libraries with SCSI Attachment	77
Determine the Drive Hardware Paths for IBM 3580 Ultrium Tape Drive, IBM Ultrium Tape Libraries with Fibre Channel Attachment	77
Create the Hardware Path Entry	77
Create the Device Specific Configuration Entries (Optional)	78
Power Off the Tape Drives	78
Install the Driver Using the CD Installation Script	78
Install Drivers Manually	79
Copy the Software to the Software Depot	79
Review the ATDD README File	80
Install ATDD	81
Configure all IBM Tape Drives/Media Changers with ATDD/ACDD.	82

Configure Selected IBM Tape Devices with ATDD/ACDD	82
Uninstalling the software	83
Other Administrative Tasks	83

Chapter 18. Install, Uninstall, and Configure the Static Version of ATDD . 85

Create the Drive Configuration File (Optional)	86
Determine the Drive Hardware Path for IBM 3580 Ultrium Tape Drive, 3581 Tape Autoloader with SCSI Attachment	86
Determine the Drive Hardware Paths for IBM Ultrium Tape Libraries with SCSI Attachment	87
Determine the Drive Hardware Paths for IBM 3580 Ultrium Tape Drive, IBM Ultrium Tape Libraries with Fibre Channel Attachment	87
Create the Hardware Path Entry	87
Create the Device Specific Configuration Entries (Optional)	88
Power Off the Tape Drives	88
Install Drivers Manually	88
Copy the Software to the Software Depot	89
Review the atdd README File	90
Install ATDD	90
Post-Configuration of IBM Medium Changer Devices	91
Adding an IBM Ultrium Device Using the Currently Installed ATDD	93
Uninstalling the Software	94
Other Administrative Tasks	94

Chapter 19. Alternate Pathing Support for Library Control Path Failover . . . 97

Configuring and Unconfiguring Alternate Pathing Support	97
Primary and Alternate Paths	98
Querying Primary and Alternate Path Configurations	98
Disable and Enable Primary and Alternate Paths	99

Chapter 20. Special Files 101

Chapter 21. Supported Configurations 103

Modifying Configuration Parameters	103
--	-----

Chapter 22. Configuration Parameter Definitions 105

Device-Specific Parameters	105
Driver-Specific Parameters	106

Chapter 23. Troubleshooting 107

Error Logging	107
Support_info Script	107
Tracing Facility	107
Atdd_d Log Daemon	108
Problems and Solutions	110

Chapter 24. Tape Utility Program (tapeutil) 113

Command Sequence Information	113
Product Requirements	113
Tapeutil Implementation	113
Hardware Requirements	113
Install tapeutil Using the CD Installation Script	114
Install tapeutil Manually	114
Copy the Software to the Software Depot	114
Review the tapeutil README File	115
Install tapeutil	115
Uninstalling tapeutil	116
Other Administrative Tasks	116
Using the tapeutil Program	116
Interactive Mode	117
Command-Line Interface	119
Service Commands	119
Basic SCSI Commands	119
Medium Changer Commands	120
CPF Commands	120
Tape Drive Commands	120
Flag Description	122

Part 5. Linux Tape and Medium Changer Device Driver 125

Chapter 25. Introduction and Product Requirements 127

Purpose	127
Data Flow	127
Product Requirements	127
Hardware Requirements for Intel Processors	128
Hardware Requirements for IBM @server pSeries Models	128
Hardware Requirements for IBM @server zSeries Models	129
Library Control Path and Tape Drive Data Path Failover Support	129
Software Requirements for Intel Processors	130
Software Requirements for IBM pSeries Models	130
Software Requirements for IBM zSeries Models	130

Chapter 26. Tape Drive, Media, and Device Driver Parameters 131

Configuration Parameters	131
Nonchangeable Parameters:	132
Autoloading	132
Capacity Scaling	132
Density Code	132
Emulate Autoloader	132
Hook Word	132
Logical Write Protect	132
Maximum Block Size	132
Minimum Block Size	132
Medium Type	132
Read SILI Bit	132
Record Space Mode	132
Volume ID for Logging	133
Write Protect	133

Changeable Parameters:	133
Block Size	133
Buffered Mode	133
Compression	133
Disable Auto Drive Dump	133
Disable SIM Logging	133
Logging (Volume Logging)	133
Maximum SCSI Transfer Length	134
Read Past Filemark	134
Rewind Immediate	134
Trace	134
Trailer Labels	135

Chapter 27. Installation and Configuration Instructions 137

Conventions Used	137
Components Created During Installation	137
Installation Procedure	138
Updating Procedure	139
Querying Installed Package	139
Verifying Installation/Updating	139
Configuring Tape and Medium Changer Devices on Intel-Compatible Systems	140
Configuring Tape and Medium Changer Devices on IBM @server pSeries Models	141
Configuring Tape and Medium Changer Devices on IBM @server zSeries Models	141
Use /etc/zfc.conf File	141
Modify the /etc/modules.conf File	142
Dynamically Attaching a Tape Device	143
Uninstall Procedure	143

Chapter 28. Configure and Run IBMtaped Daemon 145

Install IBMtaped	145
Configure IBMtaped	145
Tracing	145
Error Logging	146
Volume Logging	147
Automatically Retrieve a Drive Dump	147
Selective Tracing	148
Run IBMtaped	148

Chapter 29. Special Files 151

Special Files for the Tape Device	151
Special Files for the Medium Changer Device	151

Chapter 30. Alternate Pathing Support for Library Control Path Failover . . . 153

Configuring and Unconfiguring Alternate Pathing Support	153
Primary and Alternate Paths	154
Querying Primary and Alternate Path Configuration	155
Disable and Enable Primary and Alternate Paths	155

Chapter 31. Alternate Pathing Support for Tape Drives. 157

Automatic Failover	157
------------------------------	-----

Dynamic Load Balancing	158
Configuring and Unconfiguring Alternate Pathing Support	158
Primary and Alternate Paths	159
Querying Primary and Alternate Path Configuration	160
Disabling and Enabling Primary and Alternate Paths	160

Chapter 32. Tape Utility Program (IBMtapeutil). 161

Interactive Mode	161
Command-Line Mode	161
General Subcommands	162
disablepath "primary" number	162
enablepath "primary" number	162
inquiry [Page]	163
logpage "Page"	163
modepage "Page"	163
path	163
print "Text"	163
qrypath	163
qryversion	163
release.	164
reqsense	164
reserve	164
tur	164
Tape Subcommands	164
allow	164
append	164
asf [Count]	164
bsf [Count]	164
bsfm [Count]	164
bsr [Count]	164
compress and nocompress	164
density	165
display "Message"	165
eof [Count] and weof [Count]	165
erase	165
fsf [Count]	165
fsfm [Count].	165
fsr [Count]	165
list	165
load	165
lock	165
mtdevice	165
offline, rewoffl, and unload.	165
parms	166
prevent	166
qryinquiry	166
qrypos.	166
qrysense	166
read -d Destination [-c Count]	166
rewind and retension.	166
rtest [-b Blocksize] [-c Count] [-r Repetition]	166
rwtest [-b Blocksize] [-c Count] [-r Repetition]	167
seek [Count].	167
seod	167
setblk [Block Size]	167
setpos [Blockid]	167
status	168

sync	168
tell	168
unlock.	168
write -s Source	168
wtest [-b Blocksize] [-c Count] [-r Repetition]	168
Medium Changer Subcommands	168
allow	168
audit [Address [Count]]	169
devids.	169
elementinfo	169
exchange Source Dest1 Dest2	169
inventory.	169
mount [Slot].	169
move Source Destination	170
position Destination	170
prevent	170
unmount [Slot]	170
Service Aid Subcommands	170
dump [Filename]	170
forcedump	170
resetdrive	171
ucode [Filename]	171
Automatic Cartridge Facility Mode	171
Block Size and SCSI Transfer Size	171
Configuration Parameters	171
Capacity Scaling	171
Logical Write Protect	171
Reserve and Release Commands	171
Service Aids Commands.	172
Create Special Files	172

Chapter 33. Tape Drive Service Aids	173
Details of Tape Drive Service Aids	173
Force Drive Dump	173
Read Dump	173
Load Microcode	173
Reset Drive	173

Part 6. Solaris Tape and Medium Changer Device Driver 175

Chapter 34. Introduction and Product Requirements	177
Purpose	177
Product Requirements and Compatibility	177
Hardware Requirements.	177
Software Requirements	178
Software Requirements CPF and DPF Support	178
Data Flow	180
Software Interface to the Device Driver.	180

Chapter 35. Installation, Removal, and Configuration	181
Preventing Conflicts with Other Device Drivers	181
Installing or Updating IBMtape	182
Installation Steps	182
Removing IBMtape	188
Configuration Parameters	188
Adding or Removing Devices	191

Chapter 36. Special Files 193

Chapter 37. Alternate Pathing Support for Library Control Path 197

Configuring and Deconfiguring Alternate Pathing Support	197
Primary and Alternate Paths	198
Querying Primary and Alternate Path Configuration	198
Disable and Enable Primary and Alternate Paths	199

Chapter 38. Alternate Pathing Support for Tape Drives. 201

Automatic Failure	201
Configuring and Deconfiguring Alternate Pathing Support	202
Primary and Alternate Paths	203
Querying Primary and Alternate Path Configuration	203
Disable and Enable Primary and Alternate Paths	204

Chapter 39. Service and Diagnostic Aids 207

Functional Verification	207
Problem Determination	207
Downloading Device Microcode	208
Forcing and Storing Device Diagnostic Dump	208
Tape Monitor Daemon (tmd)	209
Tracing Facility	210
Setting the IBM_trace Level.	211
Running the Diags_info Script.	212
Tape and Medium Changer Utility Program	212
Service Commands	214
Basic SCSI Commands	214
Medium Changer Commands	215
Tape Drive Commands	215
Flag Description	216
Tapelist Utility Program (tapelist).	218

Part 7. Microsoft Windows Tape Device Drivers 219

Chapter 40. Introduction and Product Requirements 221

Windows NT	221
Hardware Requirements.	221
Software Requirements	223
Installation Notes	223
Windows 2000 and Windows Server 2003	224
Hardware Requirements.	224
Software Requirements	225
Installation Notes	225

Chapter 41. Windows NT Device Driver Management. 227

Installation Overview.	227
Installation Procedure	227
Removal Procedure	233

Manual Starting and Stopping Procedures	236
---	-----

Chapter 42. Windows 2000 and Windows Server 2003 Device Driver Management 239

Installation Overview	239
Installation Procedures	239
Installation Setup	239
Installing the Windows 200x Driver for Tape Devices	241
Installing the Windows Server 2003 Driver for Tape Devices	248
Installing the Windows 200x Driver on Medium Changers	251
Device Removal or Disable Procedure	252
Uninstalling the Device Drivers	253

Chapter 43. Alternate Pathing Support for Library Control Path Failover . . . 255

Enabling and Disabling Alternate Pathing Support	256
Querying Primary and Alternate Path Configuration	256

Chapter 44. Windows Utility Program (ntutil) 257

Calling ntutil	257
Interactive Mode	258
Batch Mode	260
Comments	261
Command Statements	261
set	271
type	271
pause	272
delay	272
system	272
symbols	272
exit	272
Symbolic Values	272
Device Driver Diagnosis Information (for Microsoft Windows NT)	275
Device Driver Diagnosis Information (for Microsoft Windows 200x)	276
Using the Debug Version	276
Restoring the Non-Debug Version	277

Part 8. 3494 Enterprise Tape Library Emulation 279

Chapter 45. 3494 Library Emulation Support for SCSI Medium Changers . 281

Overview	281
3494 Emulation Design	281
Using the 3494 API Emulation and MTLIB Program	282
SMC Library Names	282
Volume Categories	282
Asynchronous Library Operations	283

Part 9. Appendixes 285

Appendix A. Accessing Documentation and Software Online . 287

Appendix B. Verifying Proper Attachment of Your Devices 289

AIX System	289
Tape Device Attachment Test	289
Medium Changer Device Attachment Test	290
Compaq Tru64 System	290
Tape Device Attachment Test	290
Medium Changer Device Attachment Test	291
HP-UX System	291
Tape Device Attachment Test	291
Autochanger Device Attachment Test	292
Linux System	292
Tape Device Attachment Test	292
Medium Changer Device Attachment Test	293
Solaris System	294
Tape Device Attachment Test	294
Autochanger Device Attachment Test	294
Microsoft Windows System	295
Tape Device Attachment Test	295
Autochanger Device Attachment Test - Windows NT only	295

Appendix C. Managing the Microcode on the IBM Tape Drive 297

AIX System	298
Compaq Tru64 System	298
HP-UX System	298
Linux System	298
Sun Solaris System	299
Microsoft Windows System	299

Appendix D. Notices 301

Trademarks	302
----------------------	-----

Index 303

Figures

1. Attachment Array	5	25. Manual Starting and Stopping Menu	237
2. Example of an Ultrium Environment	6	26. Computer Management Console	240
3. Data Flow Process	11	27. Computer Management Console with Ultrium Device Selected	241
4. Data Path	51	28. Device Driver Properties with Reinstall Driver ... Selected	242
5. Data Flow	73	29. Install Hardware Device Drivers Dialog	243
6. TAPEUTIL Program Menu for the tape drive on HP-UX 11.00, 11i v1, and 11i v2 PCI Bus Systems	118	30. Locate Driver Files Screen with Specify a Location Selected	244
7. TAPEUTIL Program Menu for the medium changer on HP-UX 11.00, 11i v1, and 11i v2 PCI Bus Systems	118	31. Specifying the Location of the Driver	244
8. Data Flow Process	127	32. Files Needed Menu — ibmchgr.sys	245
9. Data Flow	180	33. Driver Files Search Results	245
10. TAPEUTIL Program Menu for the Tape Drive	213	34. Selecting Another Driver to Install	246
11. TAPEUTIL Program Menu for the Medium Changer	214	35. Driver Files Found screen	246
12. Select Components Menu	228	36. Completing the Upgrade Device Driver Wizard.	247
13. Start Menu	229	37. Verifying the Ultrium Device Was Installed Correctly	248
14. Start Driver List.	229	38. Hardware Update Wizard Welcome Screen	249
15. Rescan for Tape Device	229	39. Search and Installation Options	249
16. Install Driver Menu — Select Cancel	230	40. Windows Logo Testing Screen	250
17. Tape Devices Menu	231	41. Completing the Hardware Update Wizard	251
18. Have Disk Menu	232	42. Installation Application in Windows Explorer	252
19. Install Driver Menu	233	43. Base Mode	259
20. Windows NT Statement	233	44. Library Mode	260
21. Remove Driver Menu	234	45. Symbolic Values	273
22. Control Panel Selection	234	46. 3494/SMC Library Data Flow	281
23. Add/Remove Properties	235	47. LIBSMC and OS Components	282
24. Drive Removal Menu	236		

Tables

1. Ultrium Product Comparison	7	13. Components Created During IBMtapeutil Installation	138
2. Feature Codes.	12	14. Special Files for IBM Ultrium Tape Devices	151
3. Special Files for IBM 3580 Tape Device	24	15. CPF and DPF Feature Codes	179
4. Special Files	25	16. IBMtape Components	181
5. Error Description	52	17. IBMtape Install or Update	183
6. IBM Tape and Medium Changer Drivers for HP-UX (ATDD) and HP-UX System	70	18. Equipment Listing	183
7. Special Files	101	19. Equipment Listing	186
8. Device-Specific Parameter Definitions	105	20. Equipment Listing	187
9. Driver-Specific Parameter Definitions	106	21. Sample Equipment Listing	188
10. Problems, Reasons, and Solutions	110	22. IBM SCSI Tape/Medium Changer Special Files for Solaris	194
11. Problems and Solutions	111	23. Tracing Facility	210
12. Components Created During IBMtape Installation	138		

Preface

This publication provides programming reference information for IBM® Ultrium™ tape drive, medium changer, and library device drivers.

Related Information

The following sections contain lists of sources that you might need for information related to the IBM Ultrium tape drive, medium changer, and library device drivers.

IBM TotalStorage Ultrium External Tape Drive 3580

The following publications relate to the IBM TotalStorage® Ultrium External Tape Drive 3580 :

- *IBM 3580 Ultrium Tape Drive Setup, Operator, and Service Guide*, GA32-0415

IBM TotalStorage Ultrium Tape Autoloader 3581

The following publication relates to the IBM TotalStorage Ultrium Tape Autoloader 3581:

- *IBM 3581 Ultrium Tape Autoloader Setup, Operator, and Service Guide*, GA32-0412

IBM TotalStorage Ultrium Tape 2U Autoloader 3581

The following publication relates to the IBM TotalStorage Ultrium Tape 2U Autoloader 3581:

- *IBM Ultrium Tape 2U Autoloader 3581 Models L28 and F28 Setup, Operator, and Service Guide*, GA32-0470

IBM TotalStorage Ultrium Tape Library 3582

The following publications relate to the IBM TotalStorage Ultrium Tape Library 3582:

- *IBM TotalStorage Ultrium Tape Library 3582 Setup, Operator, and Service Guide*, GA32-0458

IBM TotalStorage Ultrium Scalable Tape Library 3583

The following publications relate to the IBM TotalStorage Ultrium Scalable Tape Library 3583:

- *IBM 3583 Ultrium Scalable Tape Library Setup and Operator Guide*, GA32-0411
- *IBM 3583 Ultrium Scalable Tape Library Service Guide*, GA32-0425
- *IBM Storage Area Network Data Gateway Module Setup, Operator, and Service Guide*, GA32-0436

IBM TotalStorage UltraScalable Tape Library 3584

The following publications relate to the IBM TotalStorage UltraScalable Tape Library 3584:

- *IBM 3584 UltraScalable Tape Library Planning and Operator Guide*, GA32-0408
- *IBM 3584 UltraScalable Tape Library Maintenance Information*, 19P2440

Other Publications

IBM Storage Area Network Data Gateway Installation and User's Guide, SC26-7304

Additional Information

Special Printing Instructions:

This SCSI Device Driver Manual contains different sections for each type of operating platform, for example, AIX®, HP-UX, Linux, Sun Solaris, and Windows®.

Note: When selecting the page range for the section you wish to print, note that the print page range is based on the page controls for Adobe Acrobat, not the page printed on the actual document. Enter the Adobe page numbers to print.

If you wish to print one or more separate sections of the manual, follow these steps:

1. Navigate to the beginning of the section and note the page number.
2. Navigate to the last page in the section and note that page number.
3. Select File → Print, then choose "Pages" and enter the page range for the section. Only the page range entered will print.
4. Repeat these steps to print additional sections.



Important printer note



This area indicates the pages that will actually print in your specified range of pages.

Ignore the page number appearing on the page itself when entering page ranges for your printer.

Attention: There is only one Table of Contents and one Index for this entire book. If you wish to print those items, you must repeat the process above, entering the page range of the Table of Contents and the Index page range, respectively.

The following publications contain additional information that relates to the IBM Ultrium tape drive, medium changer, and library device drivers:

- *IBM Ultrium Device Drivers: Programming Reference, GC35-0483.*
- *American National Standards Institute Small Computer System Interface X3T9.2/86-109 X3.180, X3B5/91-173C, X3B5/91-305, X3.131-199X Revision 10H, and X3T9.9/91-11 Revision 1*

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have comments or suggestions for improving this publication, you can send us comments electronically using these addresses:

- Internet: STARPUBS@us.ibm.com (or STARPUBS at us.ibm.com)
- IBMLink™ from U.S.A.: STARPUBS at SJEVM5
- IBMLink from Canada: STARPUBS at TORIBM
- IBM Mail Exchange: USIB3VVD at IBMMAIL

Summary of Changes

This summary of changes includes specific release updates to this publication.

Fifteenth Edition (May 2005)

This release includes new information related to the MTLIB SCSI Medium Changer module and additional support for Control Path Failover on Windows and HP-UX.

Fourteenth Edition (December 2004)

This release includes new information related to the Ultrium 3 Tape Drive drivers.

Thirteenth Edition (May 2004)

This release includes new information related to the IBM TotalStorage UltraScalable Tape Library 3584 Models L22 and D22 (3592 Frames) and Dynamically Loadable Kernel Module (DLKM) versions of the HP-UX drivers.

Twelfth Edition (April 2004)

This release includes new information related to the IBM TotalStorage Ultrium Tape 2U Autoloader 3581 Models L28 and F28

Eleventh Edition (October 2003)

This release includes the following new information:

- Library Path Failover information for Linux
- Information regarding IBM @server pSeries® server support

Tenth Edition (June 2003)

This release includes the following new information:

- Data Path Failover information

Ninth Edition (April 2003)

This release includes the following new information:

- IBM TotalStorage Ultrium Tape Library 3582

Eight Edition (January 2003)

This release includes the following new information:

- IBM TotalStorage UltraScalable Tape Library 3584 medium changer AIX failover
- Ultrium Generation 2 support

Fifth Edition (March 2002)

This release includes the following new information:

- Support for Linux operating systems.

- Support for new host bus adapter cards, operating system releases, and enhancements to the device drivers.
 - This release also includes changes to correct errors or omissions in the previous editions.
-

Fourth Edition (September 2001)

This release includes the following new information:

- Support for Linux operating systems.
 - Support for new host bus adapter cards, operating system releases, and enhancements to the device drivers.
 - This release also includes changes to correct errors or omissions in the previous editions.
-

Third Edition (July 2001)

This release includes the following new information:

- Support for new host bus adapter cards, operating system releases, and enhancements to the device drivers.
 - This release also includes changes to correct errors or omissions in the previous editions.
-

Second Edition (March 2001)

This release includes support for IBM @server pSeries.

Part 1. Introduction to IBM Ultrium Device Drivers

Chapter 1. Ultrium Device Drivers

This publication describes the IBM Tape and Medium Changer Device Drivers for the following devices:

- “IBM TotalStorage Ultrium External Tape Drive 3580” on page 6
- “IBM TotalStorage Ultrium Tape Autoloader 3581” on page 6
- “IBM TotalStorage Ultrium Tape 2U Autoloader 3581” on page 6
- “IBM TotalStorage Ultrium Tape Library 3582” on page 6
- “IBM TotalStorage Ultrium Scalable Tape Library 3583” on page 7
- “IBM TotalStorage UltraScalable Tape Library 3584” on page 7
- “StorageSmart by IBM Ultrium Products” on page 7

On AIX, HP-UX, Linux, Solaris, and Windows operating systems.

Purpose

The IBM Ultrium tape and medium changer device drivers are designed specifically to take advantage of the features provided by the IBM Ultrium tape drives and medium changer devices. The goal is to give applications access to the functions required for basic tape functions (such as backup and restore) and medium changer operations (such as cartridge mount and demount), as well as to the advanced functions needed by full tape management systems. Whenever possible, the driver is designed to take advantage of the device features transparent to the application.

Platform Support

Part 2, “AIX Tape and Medium Changer Device Driver,” on page 9 describes the installation and configuration of the AIX Enhanced Tape and Medium Changer Device Driver for IBM Ultrium products.

Part 4, “HP-UX Tape and Medium Changer Device Driver,” on page 67 describes the installation and configuration of the HP Enhanced Tape and Medium Changer Device Driver for IBM Ultrium products.

Part 5, “Linux Tape and Medium Changer Device Driver,” on page 125 describes the installation and configuration of the Linux Enhanced Tape and Medium Changer Device Driver for IBM Ultrium products.

Part 6, “Solaris Tape and Medium Changer Device Driver,” on page 175 describes the installation and configuration of the Solaris Tape and Medium Changer Device Driver for IBM Ultrium products, also known as IBMtape.

Part 7, “Microsoft Windows Tape Device Drivers,” on page 219 describes the installation and configuration of Microsoft® Windows-based Tape and Medium Changer Device Drivers for IBM Ultrium products.

Information in the “Appendixes” covers accessing updated drivers, microcode, and documentation online. It also addresses attachment testing of IBM Ultrium devices to the host computer.

Introduction

The IBM Ultrium product family provides an excellent solution for customers with small to large storage and performance requirements.

Figure 1 on page 5 illustrates the attachment of various Ultrium products to an open systems server.

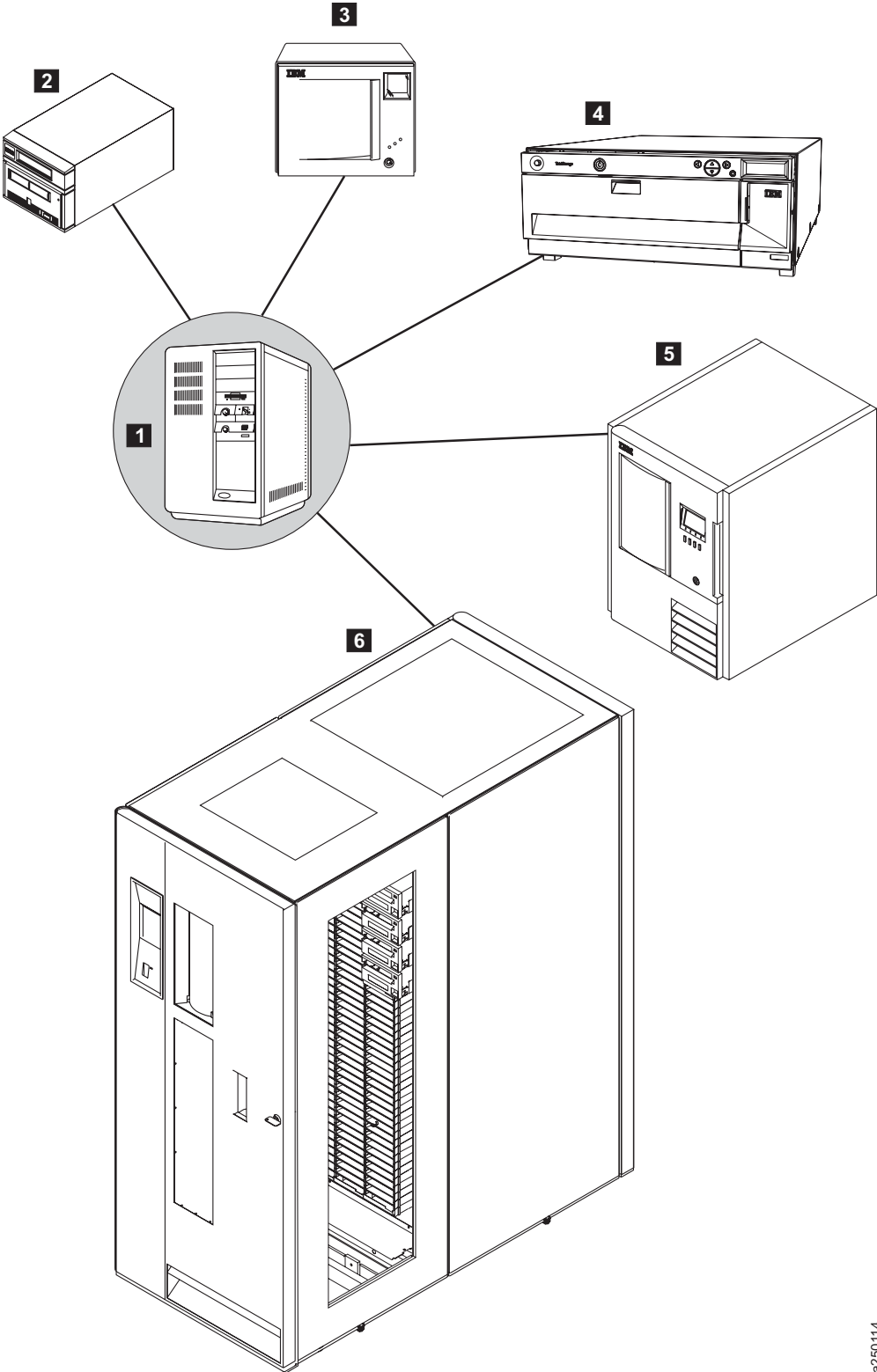


Figure 1. Attachment Array

- The following are the Ultrium Device Driver attachments shown in Figure 1:
- **1** Open Systems Server
 - **2** IBM TotalStorage Ultrium External Tape Drive 3580

a250114

Ultrium Device Driver

- **3** IBM TotalStorage Ultrium Tape Autoloader 3581
- **4** IBM TotalStorage Ultrium Tape Library 3582
- **5** IBM TotalStorage Ultrium Scalable Tape Library 3583
- **6** IBM TotalStorage UltraScalable Tape Library 3584

Figure 2 illustrates an Ultrium environment that could include an IBM 3583 Ultrium Scalable Tape Library and an IBM 3584 UltraScalable Tape Library.

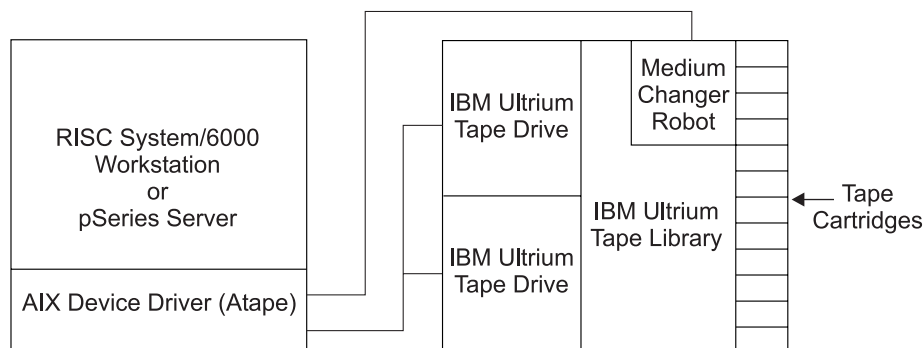


Figure 2. Example of an Ultrium Environment

IBM TotalStorage Ultrium External Tape Drive 3580

The IBM TotalStorage Ultrium External Tape Drive 3580 is a stand alone, large capacity, high performance tape drive that adheres to the Linear Tape-Open (LTO) specifications and supports the IBM Ultrium format tape. The drive supports native tape capacities of up to 200 GB with uncompressed data transfer rates of up to 35 MB per second. With both Low Voltage Differential (LVD) Ultra-2 SCSI and High Voltage Differential (HVD) Ultra SCSI attachments, this device is suitable for a variety of save, restore, and archiving requirements for PC and Open Systems platforms.

IBM TotalStorage Ultrium Tape Autoloader 3581

The IBM TotalStorage Ultrium Tape Autoloader 3581 is an external, stand-alone, or rack-mounted tape autoloader that incorporates one IBM Ultrium Tape Drive. The autoloader has seven storage slots giving the autoloader up to 1,400 GB of uncompressed data storage. The autoloader can be used with compatible software applications to automate backup/recovery or other data storage activities.

IBM TotalStorage Ultrium Tape 2U Autoloader 3581

The Ultrium Tape 2U Autoloader 3581 is an external 2U stand-alone or rack-mountable unit that incorporates a single IBM LTO Ultrium 2 tape drive. The Ultrium Tape 2U Autoloader 3581 capacity is eight tape cartridges, providing a media capacity of up to 1.6 TB (3.2 TB with 2:1 compression) data storage per unit.

The Model L28 comes with a LVD Ultra160 SCSI attachment, while the Model F28 comes with a Native Switched Fabric Fibre Channel attachment, for connection to a wide spectrum of open systems servers.

IBM TotalStorage Ultrium Tape Library 3582

The IBM TotalStorage Ultrium Tape Library 3582 is an entry tape library incorporating high-performance IBM TotalStorage Ultrium generation 2 Tape Drives for the midrange open systems environment. It can accommodate one or

two Ultrium generation 2 Tape Drives and comes standard with a one-cartridge I/O station and 23 data cartridge slots giving a native library capacity of 4.8 TB uncompressed native storage.

IBM TotalStorage Ultrium Scalable Tape Library 3583

The IBM TotalStorage Ultrium Scalable Tape Library 3583 is an automated tape library that incorporates IBM Ultrium tape drives in either a stand alone or optional rack mount configuration. Three different library models are available with storage capacities of 18 through 72 slots and one to six Ultrium tape drives. The IBM TotalStorage Ultrium Scalable Tape Library 3583 can be used for save, restore and mass storage archives where multiterrabyte capacities are required.

IBM TotalStorage UltraScalable Tape Library 3584

The IBM TotalStorage UltraScalable Tape Library 3584 is a highly scalable, automated tape library combining IBM automation technology for midrange to enterprise open systems environments. The 3584 Tape Library supports logical partitioning and can house both IBM LTO Ultrium and IBM TotalStorage Enterprise 3592 Tape Drives (in separate frames). With scalability of one to sixteen frames and up to 192 tape drives, the 3584 tape library can have an expanded available storage capacity of over 5000 TB (with 3:1 compression) using TotalStorage Enterprise 3592 Tape Drives.

For Open host attachment information regarding the IBM TotalStorage Enterprise 3592 Tape Drives, refer to the *IBM TotalStorage Tape Device Drivers Installation and User's Guide* (GC35-0154) and the *IBM TotalStorage Tape Device Drivers Programming Reference* (GC35-0346).

StorageSmart by IBM Ultrium Products

The StorageSmart™ by IBM Ultrium family of products is compatible with the IBM branded versions of the Ultrium family products. In this manual, where there is no specific mention of the StorageSmart by IBM Ultrium branded set of products, the documentation pertaining to the IBM branded family of products should be used. The following table cross references the StorageSmart by IBM Ultrium product set to the IBM Ultrium product set.

Table 1. Ultrium Product Comparison

StorageSmart by IBM Ultrium	IBM Version of Ultrium
StorageSmart by IBM Ultrium External Tape Drive TX200	IBM TotalStorage Ultrium External Tape Drive 3580
StorageSmart by Ultrium Tape Autoloader SL7	IBM TotalStorage Ultrium Tape Autoloader 3581
StorageSmart by Ultrium Scalable Tape Library SL72	IBM TotalStorage Ultrium Scalable Tape Library 3583
No equivalent in the IBM Ultrium product set	IBM TotalStorage UltraScalable Tape Library 3584

Ultrium Device Driver

Part 2. AIX Tape and Medium Changer Device Driver

Chapter 2. Introduction and Product Requirements

This chapter describes the IBM AIX Enhanced Tape and Medium Changer Device Driver for the following Ultrium products:

- IBM TotalStorage Ultrium External Tape Drive 3580
- IBM TotalStorage Ultrium Tape Autoloader 3581
- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582
- IBM TotalStorage Ultrium Scalable Tape Library 3583
- IBM TotalStorage UltraScalable Tape Library 3584

Purpose

The IBM AIX Enhanced Tape and Medium Changer Device Driver is designed specifically to take advantage of the features provided by the IBM Ultrium tape drives and medium changer devices. The goal is to give applications access to the functions required for basic tape operations, such as backup and restore, and medium changer operations, such as mount and demount the cartridges, as well as to the advanced functions needed by full tape management systems. Whenever possible, the driver is designed to take advantage of the device features transparent to the application.

Data Flow

The software described in this chapter covers the AIX Enhanced Device Driver (Atape device driver) and the interface between the application and the tape device. Figure 3 illustrates a typical Ultrium data flow process.

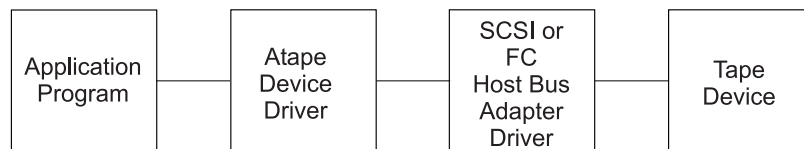


Figure 3. Data Flow Process

Product Requirements

The following software and hardware components are required to use the tape device driver.

Software Requirements

The AIX Enhanced Device Driver (Atape device driver) supports the following AIX operating system levels for operation of IBM Ultrium tape drives and automation products:

- AIX 5L™ Versions 5.1 and 5.2 on POWER-based servers

Library Control Path and Data Path Failover Support

In order to use alternate pathing support, the following feature codes are required for the associated machine type:

Table 2. Feature Codes

Machine Type	Feature Code
3582	FC 1680
3583	FC 1680
3584	FC 1680
3580 Ultrium 3	FC 1681

Hardware Requirements

The Atape device driver supports the following IBM Ultrium tape drives and automation products:

- One or more of the following IBM Ultrium tape and library devices:
 - IBM TotalStorage Ultrium External Tape Drive 3580
 - IBM TotalStorage Ultrium Tape Autoloader 3581
 - IBM TotalStorage Ultrium Tape 2U Autoloader 3581
 - IBM TotalStorage Ultrium Tape Library 3582
 - IBM TotalStorage Ultrium Scalable Tape Library 3583
 - IBM TotalStorage UltraScalable Tape Library 3584
- One or more of the following IBM RS/6000[®] or pSeries SCSI host bus adapters:
 - PCI Dual Channel Ultra-3 SCSI Adapter (LVD) (FC 6203)
 - PCI Differential Ultra SCSI Adapter (HVD) (FC 6207)
 - PCI Universal Differential Ultra SCSI Adapter (HVD) (FC 6204)
 - Integrated Ultra2 SCSI Adapter for LVD attachment (with a VHDCI connector):
 - pSeries 620 (7025 - 6F0, 6F1)
 - 7025 - F80
 - pSeries 660 (7026 - 6H0, 6H1, 6M1)
 - pSeries 640 (7026 - B80)
 - 7026 - H80, M80
 - 7044 - 170, 270
 - Integrated Ultra3 SCSI Adapter for LVD attachment (with a VHDCI connector):
 - pSeries 610 (7028 - 6C1, 6E1)
 - pSeries 630 (7028 - 6C4, 6E4)
 - pSeries 650 (7038 - 6M2)
 - pSeries 655 (7039 - 651)
 - 9112 - 265
 - PCI Dual Channel Ultra-2 SCSI Adapter (LVD) (FC 6205)
 - PCI-X Dual Ultra320 SCSI Blind Swap Adapter (FC 5710)
 - PCI-X Dual Channel Ultra 320 SCSI Adapter (FC 5712)
- One or more of the following IBM RS/6000 or pSeries FC-AL host bus adapters:
 - Gigabit Fibre Channel Adapter (PCI) (FC 6227)
 - Gigabit Fibre Channel Adapter for 64-bit PCI bus (FC 6228)
 - Two Gigabit Fibre Channel Adapter (PCI—X) (FC 6239)

Attention: Using a single Fibre Channel host bus adapter (HBA) for concurrent tape and disk operations is not recommended. Tape and disk devices require incompatible HBA settings for reliable operation and optimal performance characteristics. Under stress conditions (high I/O rates for tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems have been observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and using SAN zoning to minimize contention. IBM is focused on assuring server and storage configuration interoperability. It strongly recommends that your implementation plan includes provisions for separating disk and tape workloads.

AIX Device Driver (Atape)

Chapter 3. Tape Drive, Media, and Device Driver Parameters

This chapter describes the parameters that control the operating modes of the tape drive, media, and device driver.

Configuration Parameters

The operating parameters for the tape drive and device driver can be set and changed by the configuration procedures. The installation defaults are provided for all parameters initially. The AIX *smit* command can be used to set these parameters when configuring a device or to change these parameters. The AIX *chdev* command can also be used to change the configuration parameters.

The configuration parameters are used to set the operating mode of the tape drive and device driver when a device is opened. These parameters can be queried by an application. Some parameters can be changed temporarily during the *open* subroutine by an application, but they are always restored to the configuration values when a device is closed. The configuration parameters are:

- Autoloading
- Block size
- Compression
- Logging
- Maximum size of the log file
- Record space mode
- Rewind immediate
- Trailer labels

Autoloading

This parameter enables the autoloading feature of the device driver. It can be used when one IBM 3580 Ultrium Tape Drive is installed in an IBM 3581 Ultrium Tape Autoloader or an IBM 3583 Ultrium Scalable Tape Library. This feature allows multivolume backups (with commands such as *tar*) without prompting for a volume change.

Note: The autoloading feature is not supported on the IBM 3584 UltraScalable Tape Library and the IBM 3583 Ultrium Scalable Tape Library with more than one IBM 3580 Ultrium Tape Drive installed.

Do not enable autoloading if one of the following conditions is true:

- The device is used by an application that provides library medium changer support for the IBM 3581 or IBM 3583.
- The application is MKSYSB.
- The tapes being read were not written using the autoloading feature.

If the parameter is set to On, the tapes in the attached medium changer slots act as one large virtual tape. During a *read*, *write*, or *forward space* file operation, no end of tape is detected by the application. When the end of tape is reached, the device

AIX Device Driver (Atape)

driver automatically rewinds and unloads the tape, moves the tape back into the library, loads the next tape from the library, then continues reading or writing the next tape.

The following conditions are required to use autoloading:

- The autoloading parameter must be set to On.
- The library must be loaded with one or more tapes.
- The library mode must be set to Random mode.

The installation default is Off (no autoloading).

Block Size

This parameter specifies the block size used for read and write operations. A value of 0 is the variable block size. Any other value is a fixed block size.

The installation default is 0 (use variable length).

Compression

Hardware compression is implemented in the device hardware. This parameter turns the compression feature On and Off. If compression is enabled, the effective performance can increase based on the compressibility of the data.

The installation default is On (use compression).

Logging

This parameter turns the volume information logging On and Off. If logging is set to On, the statistical information about the device and media is saved in a log file when a tape is unloaded. If logging is set to Off, the information is not saved. This parameter has no effect on error logging because error logging is always enabled. For more information, see Chapter 12, “Device and Volume Information Logging,” on page 53.

The installation default is Off (no logging).

Maximum Size of the Log File

This parameter specifies the number of entries made before the log file starts to wrap. Each entry is approximately 2 KB (2048 bytes). After the log file starts to wrap, the number of entries remains constant. Each time a new entry is made, the oldest entry is overlaid. For more information, see Chapter 12, “Device and Volume Information Logging,” on page 53.

The installation default is 500.

Record Space Mode

This parameter specifies how the device driver operates when a forward or backward space record operation encounters a filemark. The two modes of operation are SCSI and AIX.

The SCSI mode is the default mode of operation. When a forward or backward space record operation is issued to the driver and a filemark is encountered, the device driver returns -1 and the *errno* variable is set to input/output error (EIO).

The tape is left-positioned after the filemark (the end-of-tape side of the filemark on the forward space and the beginning-of-tape side of the filemark on the backward space).

The AIX mode returns the same EIO *errno* value as the SCSI mode when a filemark is encountered, except that the tape is left-positioned before the filemark (the beginning-of-tape side of the filemark on the forward space and the end-of-tape side of the filemark on the backward space).

The installation default is SCSI mode.

Rewind Immediate

This parameter turns the immediate bit On and Off in rewind commands. If it is set to On, the rewind tape operation executes faster, but the next command takes a long time to finish unless the rewind operation is physically complete. Setting this parameter reduces the amount of time that it takes to close a device for a Rewind on Close special file.

The installation default is Off (no rewind immediate).

Trailer Labels

If this parameter is set to On, writing a record past the early warning mark on the tape is allowed. The first *write* operation to detect end of mark (EOM) fails and the *errno* variable is set to ENOSPC. No data is written during the operation. All subsequent *write* operations are allowed to continue until the physical end of the volume is reached and EIO is returned.

This parameter can also be selected using one of three device special files that allow trailer label processing. The special files are *rmtx.40*, *rmtx.41*, and *rmtx.60*, where *x* is the name of the device (for example, *rmt0.40*).

The installation default is Off (no trailer labels).

Media Parameter

The media parameter can be queried and set by the *tapeutil* application using the **Query/Set Parameters** option on the menu. This parameter cannot be set or changed by the configuration procedures. The media parameter is:

- Volume ID for logging

This parameter is the volume ID of the currently loaded tape. It is used in the log file entry (if volume logging is active) to identify the entry with a particular volume. The device driver sets the volume ID to UNKNOWN initially and when the tape is unloaded.

AIX Device Driver (Atape)

Chapter 4. Installation and Configuration Instructions

The recommended procedure for installing and configuring the device driver is to use the installation script provided on the Open Systems Device Driver CD. However, before this is done, you need to physically attach the device to the server. Consult the appropriate manuals for instructions on physically attaching the tape device to the server.

Instructions for uninstalling the device driver are outlined in “Uninstallation Procedure” on page 21.

Attention: At the end of the installation procedure, the *installp* facility will automatically run the AIX *bosboot* command to update the boot record with the newly installed Atape files. When the *bosboot* command completes, the following messages will be displayed:

```
0503-292 This update will not fully take effect until after a system reboot.
installp: bosboot process completed.
```

This message is referring to the updates to the boot record only. If the installation summary shows that the Atape driver was installed successfully, it is **not** necessary to reboot the machine at this time.

If the installation summary shows that the install failed, you should reboot the machine and attempt to install the Atape driver a second time.

Installation Procedure

For information on obtaining the latest version of device drivers and the latest documentation, see Appendix A, “Accessing Documentation and Software Online,” on page 287.

Preinstallation Considerations

Before proceeding with the installation, verify the following items:

1. The tape device is properly functioning, properly attached to the server, and is powered up.
2. You have logged onto the server on an account that has root authority.
3. You have a command shell window open on the server to perform the installation procedure.
4. Make sure the current path is defined in the command shell PATH variable. This can be accomplished in the korn shell using the following command:

```
EXPORT PATH=.:$PATH
```
5. If the tape device was configured previously by another device driver (not Atape), remove any existing device definitions for it. For example, `'rmdev -l rmt1 -d'`.

Installation Procedure

Enter the following command to list the currently installed *Atape.driver* version:

```
lspp -l Atape.driver
```

If you have the Tape Device Drivers and SMI-S Agent CD, use the following instructions to install and configure the device driver:

1. Place the CD into the CD-ROM drive on your AIX system.
2. Mount the CD over an empty directory. For example, if your CD-ROM drive is defined at */dev/cd0* and you have an empty directory at */cdrom*, issue the following command to mount the CD:

```
mount -frv cdrfs /dev/cd0 /cdrom
```

You can create an empty directory using the *mkdir* command, for example:

```
mkdir /cdrom
```

Subsequent instructions assume that you mounted the CD at mount point */cdrom*.

3. Enter the following command:

```
cd /cdrom/Drivers/AIX
```
4. Consult the *Atape.Readme* file on the CD for any important information pertaining to the device driver. Information in this file takes precedence over information in the manual.
5. Execute the *install_atape* script. This script uninstalls any previous versions of Atape, installs and commits the latest version of Atape, then runs *cfgmgr* to define your devices.
6. Enter the following commands:

```
cd /  
umount /cdrom
```
7. Remove the CD from the CD-ROM drive and store it in a safe place.

Configuring Tape and Medium Changer Devices

After the driver software is installed and a tape device is connected to the adapter, the device can be configured and made available for use. Access to the device is not provided until the device is configured.

Note: If the tape device was configured previously by another SCSI device driver, such as OST (Other SCSI Tape), issue the following command to remove the device definition before performing the following steps:

```
rmdev -l [device]
```

Configure a tape device using **one** of the following procedures:

- Enter the following command with no parameters:

```
cfgmgr
```

The command configures all devices automatically (including any new tape or medium changer devices).

or

- Power Off your subsystem and reboot the system to configure it automatically and make available any new tape or medium changer devices on the system.

Deconfiguring Tape Devices

In the following examples, replace the letter *n* with the appropriate number for the chosen device.

Deconfigure the tape device using **one** of the following procedures:

- The first method leaves the tape device defined in the configuration database. It is similar to bringing the device Offline (not in use).

Enter the following command to bring the `/dev/rmtn` tape device Offline but leave it defined in the device database:

```
rmdev -l rmtn
```

or

- The second method brings the tape device Offline and removes its definition from the device database.

Enter the following command:

```
rmdev -l rmtn -d
```

The device driver is not unloaded from the kernel until the last device is deconfigured.

Deconfiguring Medium Changer Devices

In the following examples, replace the letter *n* with the appropriate number for the chosen device.

Deconfigure the medium changer device using **one** of the following procedures:

- The first method leaves the device defined in the configuration database. It is similar to bringing the device Offline.

Enter the following command to bring the `/dev/smcn` medium changer device Offline but leave it defined in the device database:

```
rmdev -l smcn
```

or

- The second method brings the medium changer device Offline and removes its definition from the device database.

Enter the following command:

```
rmdev -l smcn -d
```

The device driver is not unloaded from the kernel until the last device is deconfigured.

Uninstallation Procedure

All tape devices that use the Atape driver must be closed and cannot be in use when Atape is uninstalled or the uninstall will fail.

You can uninstall the Atape device driver using the `smit` command menu to uninstall software and selecting *Atape.driver* or using the `installp` command:

```
installp -u Atape.driver
```

AIX Device Driver (Atape)

Chapter 5. Special Files

After the driver is installed and a tape device is configured and made available for use, access is provided through the special files. These special files, which consist of the standard AIX special files for tape devices (along with other files unique to the Atape driver), are in the */dev* directory.

Special Files for Tape Devices

Each tape device has a set of special files that provides access to the same physical drive but to different types of functions. As shown in Table 3, in addition to the tape special files, a special file is provided for tape devices that allows access to the medium changer as a separate device.

Note: The asterisk (*) represents a number assigned to a particular device, such as *rmt0*.

For tape drives with attached medium changer devices, the *rmt*.smc* special file provides a separate path for issuing commands to the medium changer. When this special file is opened, the application can view the medium changer as a separate device.

Both this special file and the *rmt** special file can be opened at the same time. The file descriptor that results from opening the *rmt*.smc* special file does not support the following operations:

- Read
- Write
- Open in diagnostic mode
- Commands designed for a tape device

If a tape drive has a medium changer device attached, then all operations (including the medium changer operations) are supported through the interface to the *rmt** special file.

AIX Device Driver (Atape)

Table 3. Special Files for IBM 3580 Tape Device

Special File Name	Rewind on Close	Retension on Open	Bytes per Inch	Trailer Label	Unload on Close
/dev/rmt*	Yes	No	N/A	No	No
/dev/rmt*.1	No	No	N/A	No	No
/dev/rmt*.2	Yes	Yes	N/A	No	No
/dev/rmt*.3	No	Yes	N/A	No	No
/dev/rmt*.4	Yes	No	N/A	No	No
/dev/rmt*.5	No	No	N/A	No	No
/dev/rmt*.6	Yes	Yes	N/A	No	No
/dev/rmt*.7	No	Yes	N/A	No	No
/dev/rmt*.10	No	No	N/A	No	No
/dev/rmt*.20	Yes	No	N/A	No	Yes
/dev/rmt*.40	Yes	No	N/A	Yes	No
/dev/rmt*.41	No	No	N/A	Yes	No
/dev/rmt*.60	Yes	No	N/A	Yes	Yes
/dev/rmt*.null	Yes	No	N/A	No	No
/dev/rmt*.smc	N/A	N/A	N/A	N/A	N/A

Notes:

1. The Rewind on Close special files write filemarks under certain conditions before rewinding. See the *IBM Ultrium Device Drivers: Programming Reference*.
2. The Retension on Open special files rewind the tape only on *open*. Retensioning is not performed because these tape products perform the retension operation automatically when needed.
3. The Bytes per Inch options are ignored for the tape devices that this driver supports. The density selection is automatic.
4. The *rmt*.null* file is a pseudo device similar to the */dev/null* AIX special file. The *ioctl* calls can be issued to this file without a real device attached to it, and the device driver will return a successful completion. Read and write system calls will return the requested number of bytes. This file can be used for application development or debugging problems.
5. The *rmt*.smc* file can be opened independently of the other tape special files.
6. The *rmt*.10* file bypasses normal close processing, and the tape is left at the current position.

Special Files for Medium Changer Devices

After the driver is installed and a medium changer device is configured and made available for use, access to the robotic device is provided through the *smc** special file in the */dev* directory.

Table 4 shows the attributes of the special file. The asterisk (*) represents a number assigned to a particular device, such as *smc0*. The term *smc* is used for a SCSI medium changer device. The *smc** special file provides a path for issuing commands to control the medium changer robotic device. This same terminology is extended for medium changers, which are attached by way of a fibre channel.

Table 4. Special Files

Special File Name	Description
<i>/dev/smc*</i>	Access to the medium changer robotic device
<i>/dev/smc*.null</i>	Pseudo medium changer device
<p>Note: The <i>smc*.null</i> file is a pseudo device similar to the <i>/dev/null</i> AIX special file. The commands can be issued to this file without a real device attached to it, and the device driver will return a successful completion. This file can be used for application development or debugging problems.</p>	

The file descriptor that results from opening the *smc* special file does not support the following operations:

- Read
- Write
- Commands designed for a tape device

AIX Device Driver (Atape)

Chapter 6. Alternate Pathing Support for Library Control Path Failover

Note: The library control path failover feature code must be installed prior to enabling the alternate pathing support in the Atape device driver. Refer to “Library Control Path and Data Path Failover Support” on page 11 for what feature code is required for your machine type.

The Atape device driver alternate pathing support will configure multiple physical control paths to the same logical library within the device driver and provide automatic failover to an alternate control path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) in an RS/6000 or pSeries host that are connected to a library with two drive control ports enabled. The first HBA is connected to the first control port drive, and the second HBA is connected to the second control port drive. This connection provides two physical control paths to the same library for redundancy if one path from an HBA to the library fails.

When the RS/6000 or pSeries is booted or `cfgmgr` is run, each HBA detects a control port to the same library, and two logical devices will be configured (for example, `smc0` and `smc1`). Each logical device is a physical path to the same library. An application can open and use only one logical device at a time, either `smc0` or `smc1`, because they represent the same physical device.

Without the Atape alternate pathing support, if an application opens `smc0` and a permanent path error occurs (because of an HBA, cable, or drive control port failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (`smc1`), but the application has to be restarted from the beginning.

When the alternate pathing support is enabled on both `smc0` and `smc1`, the device driver configures them internally as a single device with multiple paths. The application can still open and use only one logical device at a time (either `smc0` or `smc1`). If an application opens `smc0` and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (`smc1`). If successful, the current operation continues on the alternate path without interrupting the application.

Configuring and Unconfiguring Alternate Pathing Support

Alternate pathing support is not enabled automatically when the device driver is installed. It must be configured initially on each logical device after installation. When alternate pathing support is enabled for a logical device, it remains set until the device is deleted or the support is unconfigured. The alternate pathing setting is retained even if the system is rebooted.

To enable or disable the support on a single logical device, use the smit menu to Change/Show Characteristics of a Tape Drive, select the logical device to change such as smc0, smc1, and so on, then select Yes or No for Enable Alternate Pathing Support. The support can also be enabled or disabled using the chdev command, for example:

```
chdev -l smc0 -aalt_pathing=yes
chdev -l smc1 -aalt_pathing=yes
chdev -l smc0 -aalt_pathing=no
chdev -l smc1 -aalt_pathing=no
```

Primary and Alternate Paths

When the device driver configures a logical device with alternate pathing support enabled, the first device configured always becomes the primary path. On SCSI attached devices, -P is appended to the location field and on Fibre attached devices -PRI is appended to the location field of the device.

When a second logical device is configured with alternate pathing support enabled for the same physical device, it configures as an alternate path. On SCSI attached devices, -A is appended to the location field and on Fibre attached devices -ALT is appended to the location field of the device. A third logical device is also configured as an alternate path with either -A or -ALT appended, and so on. The device driver supports up to 16 physical paths for a single device.

For example, if an smc0 is configured first, then an smc1, and an smc2, the lsdev -Cc tape command location field would look something like the following:

```
smc0 Available 20-60-01-PRI   IBM 3584 Library (FCP)
smc1 Available 30-68-01-ALT   IBM 3584 Library (FCP)
smc2 Available 30-68-01-ALT   IBM 3584 Library (FCP)
```

The labeling of a logical device as either a primary or alternate path is for information only, in order to:

1. Be able to identify the actual number of physical devices configured on the system and a specific logical device associated with them. There will be only one logical device labeled as the primary path for each physical device. However, there can be many (multiple) logical devices labeled as an alternate path for the same devices.
2. Provide information about which logical devices configured on the system have alternate pathing support enabled.

Querying Primary and Alternate Path Configuration

You can display the primary and alternate path configuration for all devices with the lsdev command. There can be two or more logical devices configured for a single physical device, but the first device configured is labeled the primary device. All other logical devices configured after the first device are labeled as alternate devices. To see this, run the lsdev -Cc tape command and look at the location field

in the data. By running `lsdev -Cc tape | grep P`, for example, you can easily determine how many physical devices are configured with alternate pathing support.

You can display the primary and alternate path configuration for any device by running:

```
tapeutil -f/dev/smcx path (where smcx is the logical name of any device)
```

This command shows specific information for the primary path and all alternate paths, such as the logical name of the device, SCSI IDs, the current enabled status, and how many paths are configured for the device.

Configuring and Unconfiguring Primary and Alternate Devices

Logical devices configured as alternate paths can be unconfigured and reconfigured at any time after the initial configuration is run. Unconfiguring an alternate path device removes that device from the primary device path list, removes the `-A` or `-ALT` appended to the location field, and changes the device to the Defined state. The primary and any other alternate devices are still available.

Likewise, configuring a new alternate path device or reconfiguring an existing one in the Defined state adds that device to the primary device path list, appends `-A` or `-ALT` to the location field, and makes the device available.

Logical devices that are configured as primary paths can also be unconfigured and reconfigured at any time after initial configuration is run. However, the operation is different for a primary device. When a primary device is unconfigured, the following events occur:

1. All alternate devices are unconfigured as described previously.
2. The primary device is unconfigured.
3. The `-P` or `-PRI` appended to the location field is removed.
4. The device is changed to the Defined state.
5. All alternate devices that were unconfigured are reconfigured. The first device that is reconfigured becomes the new primary device. All remaining alternate devices are reconfigured as alternate paths.

These methods provide the ability to unconfigure and reconfigure physical devices on the system when device connections or addressing changes are made.

AIX Device Driver (Atape)

Chapter 7. Alternate Pathing Support for Tape Drives

The Atape device driver alternate pathing support will configure multiple physical paths to the same device within the device driver and provides two basic functions:

1. Automatic failover to an alternate physical path when a permanent error occurs on one path.
2. Dynamic load balancing for devices using multiple Host Bus Adapters (HBA).

Automatic Failover

The automatic failover support provides error recovery on an alternate path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) in an RS/6000 or pSeries host that is connected through a switch to the tape drive. This connection provides two physical paths to same tape drive for redundancy if one path from an HBA to the drive fails.

When the RS/6000 or pSeries is booted or *cfgmgr* is run, each HBA detects a tape drive, and two logical devices will be configured. For example, *rmt0* and *rmt1*. Each logical device is a physical path to the same tape drive. A backup and restore application can open and use only one logical device at a time, either *rmt0* or *rmt1*, because they represent the same physical device.

Without the Atape alternate pathing support, if an application opens *rmt0* and a permanent path error occurs (because of an HBA or cable failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (*rmt1*), but the application has to be restarted from the beginning. A long backup or restore operation could have been in progress when the path error occurred. Sometimes manual failover can require operator intervention to reset the drive because a SCSI Reservation could still exist on the failing HBA path.

When the alternate pathing support is enabled on both *rmt0* and *rmt1*, the device driver configures them internally as a single device with multiple paths. The application can still open and use only one logical device at a time (either *rmt0* or *rmt1*). If an application opens *rmt0* and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (*rmt1*). If successful, the current operation continues on the alternate path without interrupting the application. The failover error recovery first restores the previous device state, SCSI Reservation, and tape position, then retries the failing operation.

Notes:

1. This function is supported on IBM 3580 Ultrium II Fibre Channel tape drives only. This function is not supported for devices that are attached through an IBM San Data Gateway.
2. The AIX operating system only supports a static configuration of devices, which also applies to the Alternate Pathing and Failover Support. When devices are initially configured at a specific SCSI ID and physical connection (drive port, host bus adapter, and switch number/port, if applicable) and in the

Available state, changing the physical device address/connection without either rebooting or unconfiguring and reconfiguring the devices has unpredictable results and is not supported.

Dynamic Load Balancing

The dynamic load balancing support optimizes resources for devices that have physical connections to multiple Host Bus Adapters (HBA) in the same machine. When an application opens a device that has multiple HBA paths configured, the device driver determines which path has the HBA with the lowest usage, and assigns that path to the application. When another application opens a different device with multiple HBA paths, the device driver again determines the path with the lowest HBA usage and assigns that path to the second application. The device driver will update the usage on the HBA assigned to the application when the device is closed. Dynamic load balancing will use all Host Bus Adapters whenever possible and balance the load between them to optimize the resources in the machine.

For example, consider a machine with two Host Bus Adapters, HBA1 and HBA2, with multiple tape drives attached. Each tape drive is connected to both HBA1 and HBA2. Initially, there are no tape drives currently in use. When the first application opens a tape drive for use, the device driver will assign the application to use HBA1. When a second application opens a tape drive for use, the device driver will assign the second application to use HBA2. A third application would be assigned to HBA1 and a fourth application would be assigned to HBA2. There would be two applications using HBA1 and two applications using HBA2.

If the first application finishes and closes the device, there would now be one application using HBA1 and two applications using HBA2. When the next application opens a tape drive, it would be assigned to HBA1, so again there would be two applications using HBA1 and two applications using HBA2. Likewise, if the second application finishes and closes the device, HBA2 would have one application using it and the next application that opens a tape drive would be assigned to HBA2.

The dynamic load balancing support is independent from the automatic failover support. Regardless of the path assigned initially for load balancing, if that path fails the automatic failover support will attempt recovery on the next available path.

Installing Feature Code 1681 Data Path Failover License Key

Alternate pathing support for IBM 3580 Ultrium 3 tape drives requires the Feature Code 1681 DPF license key. Use the following command line script to query, add, or delete license keys for this feature before enabling the alternate pathing feature as described below. The key is a 16-digit hexadecimal value, for example, 1234567890abcdef.

All key values “A-F” should be entered in lowercase letters as “a-f.”

- Query installed keys: `dpf_keys`
- Install a license key: `dpf_keys -a key`
- Delete a license key: `dpf_keys -d key`

Configuring and Unconfiguring Alternate Pathing Support

Alternate pathing support is not enabled automatically when the device driver is installed. It must be configured initially on each logical device after installation. When alternate pathing support is enabled for a logical device, it remains set until the device is deleted or the support is unconfigured. The alternate pathing setting is retained even if the system is rebooted.

Alternate pathing support can be enabled on all configured devices at one time, or it can be enabled or disabled selectively by logical device. It might be desirable at times to configure some, but not all, logical paths to a device with the support enabled.

To enable the support globally on all currently configured devices, run the command:

```
/usr/lpp/Atape/instAtape -a
```

This will unconfigure all devices that have alternate pathing set to No, and will reconfigure all devices, setting alternate pathing to Yes.

To enable or disable the support on a single logical device, use the *smit* menu to **Change/Show Characteristics of a Tape Drive**, then select Yes or No for **Enable Alternate Pathing Support**. The support can also be enabled or disabled using the *chdev* command, for example:

```
chdev -l rmt0 -aalt_pathing=yes
```

```
chdev -l rmt0 -aalt_pathing=no
```

Primary and Alternate Paths

When the device driver configures a logical device with alternate pathing support enabled, the first device configured always becomes the primary path and PRI is appended to the location field of the device. When a second logical device is configured with alternate pathing support enabled for the same physical device, it configures as an alternate path and ALT is appended to the location field. A third logical device is configured as the next alternate path with ALT appended, and so on. The device driver supports up to 16 physical paths for a single device.

For example, if *rmt0* is configured first, then *rmt1*, the *lsdev -Cc* tape command output will be similar to the following:

```
rmt0 Available 20-60-01-PRI IBM 3580 Ultrium Tape Drive (FCP)
rmt1 Available 30-68-01-ALT IBM 3580 Ultrium Tape Drive (FCP)
```

If *rmt1* is configured first, then *rmt0*, the command output will be similar to the following:

```
rmt0 Available 20-60-01-ALT IBM 3580 Ultrium Tape Drive (FCP)
rmt1 Available 30-68-01-PRI IBM 3580 Ultrium Tape Drive (FCP)
```

The labeling of a logical device as either a primary or alternate path is for information only, in order to:

1. Be able to identify the actual number of physical devices configured on the system and a specific logical device associated with them. There will be only one logical device labeled the primary path for each physical device. However, there can be many (multiple) logical devices labeled as an alternate path for the same devices.

2. Provide information about which logical devices configured on the system have alternate pathing support enabled.

Querying Primary and Alternate Path Configuration

You can display the primary and alternate path configuration for all devices with the *lsdev* command. There can be two or more logical devices configured for a single physical device, but the first device configured is labeled the primary device. All other logical devices configured after the first device are labeled as alternate devices. To see this, run the *lsdev -Cc tape* command and look at the location field in the data. By running *lsdev -Cc tape | grep PRI*, for example, you can easily determine how many physical devices on the RS/6000 or pSeries server are configured with alternate pathing support. You can display the primary and alternate path configuration for a single device by running the *tapeutil -f/dev/rmtx path* command (where *rmtx* is the logical name of any device).

This command shows specific information for the primary path and all alternate paths, such as the logical name of the device, SCSI IDs, the current enabled status, and how many paths are configured for the device.

Configuring and Unconfiguring Primary and Alternate Devices

Logical devices configured as alternate paths can be unconfigured and reconfigured at any time after the initial configuration is run. Unconfiguring an alternate path device removes that device from the primary device path list, removes the *ALT* appended to the location field, and changes the device to the Defined state. The primary and any other alternate devices are still available. Likewise, configuring a new alternate path device or reconfiguring an existing one in the Defined state adds that device to the primary device path list, appends *ALT* to the location field, and makes the device available.

Logical devices that are configured as primary paths can also be unconfigured and reconfigured at any time after initial configuration is run. However, the operation is different for a primary device. When a primary device is unconfigured, the following events occur:

1. All alternate devices are unconfigured as described previously.
2. The primary device is unconfigured.
3. The *PRI* appended to the location field is removed.
4. The device is changed to the Defined state.
5. All alternate devices that were unconfigured are reconfigured. The first device that is reconfigured becomes the new primary device. All remaining alternate devices are reconfigured as alternate paths.

These methods provide the ability to unconfigure and reconfigure physical devices on the system when device connections or addressing changes are made.

Chapter 8. Using the Dump Support

Dump support is provided through the dump entry point in the driver. See the appropriate AIX manuals for a description of how to use the dump devices and how to read the dump data. Review the *sysdumpdev* and *sysdumpstart* commands.

To list the current dump devices, enter the following command:

```
sysdumpdev -l
```

To establish the *rmt1* tape device as a secondary dump device, enter the following command:

```
sysdumpdev -s /dev/rmt1
```

To perform a dump operation, use the *sysdumpstart* command. To send the dump data to the secondary dump device, enter the following command:

```
sysdumpstart -s
```

Note: This command stops the system. Use the *sync* command to ensure that the cache is flushed before issuing the *sysdumpstart -s* command.

To list the last dump data, enter the following command:

```
sysdumpdev -z
```

After the dump data is placed on the tape, copy it to a file on the disk before using the *crash* command to process it. For example:

```
dd if=/dev/rmt1 of=tapedump1 ibs=4096 obs=512  
crash tapedump1
```

Note: The *ibs* value is the input block size.

If the block size of the tape device is larger than the block size sent during the dump process, the dump operation fails. Set the block size to zero on the tape device and experiment with the *ibs* value for the *dd* command.

AIX Device Driver (Atape)

Chapter 9. Tape Utility Program (*tapeutil*)

Installed with the device driver is a tape utility program (*tapeutil*) that exercises or tests the functions of the tape device and the device driver. It also performs basic tape and medium changer operations. The tape utility program provides two versions (the interactive interface and the AIX command-line interface with syntax similar to the *tctl* and *mt* commands).

The C source code for the *tapeutil.c* program can be found in the */usr/lpp/Atape/samples* directory. The program contains a sample of the interface to the device driver and the input/output control (*ioctl*) commands supported by the device driver.

Interactive Interface

The interactive interface of the tape utility program can be called from the AIX command line using the *tapeutil* command. A list of general subcommands, medium changer subcommands, and tape subcommands is displayed. You must open a device before using these commands and operations (except for the tape drive service aids).

To open a device:

1. Select **Open a Device** from General Commands.
2. Enter the name of the device special file. Use any special file that exists for the device, for example, */dev/rmt0*, */dev/rmt0.1*, */dev/rmt1.smc*, or */dev/smc0*.
3. Enter the *Read/Write*, *Read Only*, *Write Only*, or *Append* mode to open a device. These modes apply to the tape devices only.

After you open a device, select a command using the appropriate number for the command from the menu. Some commands require additional information after they are selected. If an error occurs for the command, the error number, the error text, and the device sense data (if applicable) are displayed.

Command-Line Interface

The command-line interface of the tape utility program (*tapeutil*) has a syntax similar to the AIX *tctl* and *mt* commands and provides the same basic tape commands. The program also supports tape device, device driver, SCSI, and medium changer subcommands that use the additional functions of the tape device and device driver.

You can call the *tapeutil* command from the AIX command line or from within a shell script. If you enter the *tapeutil* command without any arguments, the interactive interface is called.

The syntax for the command-line interface of the tape utility program is:

```
tapeutil -f Device Subcommand [Subcommand ...]
```

Notes:

1. The *Device* is the name of the device special file (for example, */dev/rmt0*).
2. The *Subcommand* is any valid command for the device.

Multiple subcommands can be combined in a single command to perform more than one operation. The subcommands are processed one at a time in the order specified on the command line. For a list of subcommands and syntax, enter the `tapeutil ?` command on the AIX command line.

General Subcommands

The following general subcommands are available for the tape and medium changer devices:

- **devinfo**

This subcommand displays the device information returned from the `IOCINFO ioctl` command.

- **fuser**

This subcommand is similar to the AIX `fuser` command. If the device special file is already open by a process, it displays the process ID; otherwise, it indicates the device special file is not currently open.

- **inquiry [Page]**

This subcommand issues the SCSI Inquiry command to the device for either standard inquiry data if the `page` parameter is omitted, or for the specified page, and displays the inquiry data. The `page` parameter must be specified as a hex value. For example:

```
# Get standard inquiry data
tapeutil -f/dev/rmt0 inquiry
```

```
# Get inquiry page x'83'
tapeutil -f/dev/rmt0 inquiry 83
```

- **kill**

This subcommand kills a currently active process on a device. This subcommand should be used when the standard Ctrl-C or AIX `kill` command methods cannot terminate the process. For example:

```
# Kill any active process on rmt0
tapeutil -f/dev/rmt0 kill
```

- **logpage “Page”**

This subcommand issues the SCSI Log Sense command to the device for the specified page and displays the `log sense` data. The `page` parameter must be specified as a hex value. For example:

```
# Get log page x'2E'
tapeutil -f/dev/rmt0 logpage 2e
```

- **loop [Count]**

This subcommand loops all subsequent subcommands continuously or a number of times if the `Count` parameter is specified. Also see the `sleep` subcommand.

Examples:

```
# Continuously loop inquiry commands with a 2-second delay
tapeutil -f/dev/rmt0 loop inquiry sleep 2
```

```
# Issue 3 Test Unit Ready commands
tapeutil -f/dev/rmt0 loop 3 tur
```

- **modepage “Page”**

This subcommand issues the SCSI Mode Sense command to the device for the specified page and displays the `mode sense` data. The `page` parameter must be specified as a hex value. For example:

```
# Get mode page x'1D'
tapeutil -f/dev/rmt0 modepage 1d
```

- **passthru**

This subcommand opens the device special file using the SC_PASSTHRU mode. This mode bypasses normal open and close processing, and no SCSI commands are issued to the device during *open* or *close*.

- **path**

This subcommand displays information about the device and SCSI paths, such as logical parent, SCSI IDs, and the status of the SCSI paths for the primary path and all alternate paths that are configured. The output from this subcommand is similar to the *qrypath* subcommand, except more than one alternate path is supported by this subcommand. For example:

```
# Display path information
tapeutil -f/dev/rmt0 path
```

- **print "Text"**

This subcommand prints the associated text to standard output. It can be used at any time to display the progress of the subcommands. For example:

```
# Set volume id, erase current tape and backup myfile.tar
tapeutil -f/dev/rmt0 volid "My Volume"      \
                    rewind                  \
                    erase                   \
                    print "Writing myfile.tar" \
                    write -s myfile.tar
```

- **qrypath**

This subcommand displays information about the device and SCSI paths, such as logical parent, SCSI IDs, and status of the SCSI paths.

- **release**

This subcommand releases a device explicitly and makes it available for other hosts. See “Reserve and Release Commands” on page 46 for more information.

- **reqsense**

This subcommand issues the SCSI Request Sense command to the device and displays the sense data.

- **reserve**

This subcommand reserves a device explicitly. See “Reserve and Release Commands” on page 46 for more information.

- **reset**

This subcommand opens the device special file using SC_FORCED_OPEN mode and causes a bus device reset to be sent to the device.

Note: You must have *root* authority to use this subcommand.

- **sleep Seconds**

This subcommand causes the *tapeutil* program to sleep the specified seconds before executing the next subcommand. Also see the *loop* subcommand.

Examples:

```
# Issue 2 inquiry commands with a 1-second delay between commands
tapeutil -f/dev/rmt0 inquiry sleep 1 inquiry
```

```
# Loop inquiry commands with a 2-second delay between commands
tapeutil -f/dev/rmt0 loop inquiry sleep 2
```

- **tur**

This subcommand issues the SCSI Test Unit Ready command to the device.

- **vpd**

This subcommand obtains the vital product data (VPD) from a SCSI tape device. It opens the device special file in SC_DIAGNOSTIC mode and uses the SCSI pass-through *ioctl* command to obtain the inquiry data from the device.

Medium Changer Subcommands

The following medium changer subcommands are available for the integrated and independent medium changer devices:

- **allow**

This subcommand allows medium removal by an operator. It is normally used after the *prevent* subcommand to restore the device to the default state.

- **audit [Address[Count]]**

This subcommand with no parameters issues the SCSI Initialize Element Status command to the device.

Using the optional parameters *Address* and *Count* issues the SCSI Initialize Element Status With Range command to the device. The *Address* parameter specifies the starting element address and the *Count* parameter, if used, specifies the number of elements to initialize. If *Count* is omitted, it defaults to 1. For example:

```
# Initialize all elements
tapeutil -f/dev/smc0 audit

# Initialize element 32
tapeutil -f/dev/smc0 audit 32

# Initialize elements 36 to 40
tapeutil -f/dev/smc0 audit 36 5
```

- **devids**

This subcommand issues the SCSI Read Element Status command to the device with the *read device ID* option for all drive elements and displays the element status information, which includes the *device ID* field.

- **elementinfo**

This subcommand displays the information returned from the `SMCIOC_ELEMENT_INFO ioctl` command that contains the number and addresses of each element type.

- **exchange “Source” “Dest1” “Dest2”**

This subcommand issues the SCSI Exchange Medium command to the device using the *Source*, *Dest1*, and *Dest2* addresses specified. This command performs the equivalent function of two Move Medium commands that first move the cartridge from the element address specified by the *Dest1* parameter to the element address specified by the *Dest2* parameter, then move the cartridge from the element address specified by the *source* parameter to the element address specified by the *Dest1* parameter. For example:

```
# Exchange cartridge in slot 34 with cartridge in drive address 16
# and return that cartridge to slot 40 (drive must be unloaded first)
tapeutil -f/dev/smc0 exchange 34 16 40

# Use move medium commands to perform exchange
tapeutil -f/dev/smc0 move 16 40 move 34 16
```

- **inventory [-i | -v Volid]**

This subcommand with no parameters issues the SCSI Read Element Status command for each element type and displays the element status information. If the optional *-i* parameter is used, then only the import/export element status information will be returned. If the optional *-v* parameter is used, then only the element status information for the specified *Volid* if found will be returned.

- **mount [Slot]**

This subcommand mounts a tape from the specified *slot* into the drive or from the first full *slot* into the drive if the *slot* is omitted. For example:

```
# Mount cartridge from slot 3
tapeutil -f/dev/smc0 mount 3
```

```
# Mount cartridge from first full slot
tapeutil -f/dev/smc0 mount
```

- **move “Source” “Destination”**

This subcommand issues the SCSI Move Medium command using the *source* and *destination* addresses specified. The element addresses can be obtained using the *elementinfo* subcommand. For example:

```
# Get slot and drive addresses
tapeutil -f/dev/smc0 elementinfo
```

```
# Move cartridge in slot 20 to drive at address 16
tapeutil -f/dev/smc0 move 20 16
```

- **position “Destination”**

This subcommand issues the SCSI Position to Element command using the *destination* specified. For example:

```
# Position to slot at address 20
tapeutil -f/dev/smc0 position 20
```

- **prevent**

This subcommand prevents medium removal by an operator until the *allow* subcommand is issued or the device is reset.

- **unmount [Slot]**

This subcommand moves a tape from the drive to the specified *slot* or the first empty one if the *slot* is omitted. The tape is rewound and unloaded automatically from the drive first when this command is issued to the tape device special file. For example:

```
# Move tape from drive to slot 4 (tape is already unloaded)
tapeutil -f/dev/smc0 unmount 4
```

```
# Unload tape and move to the first empty slot
tapeutil -f/dev/rmt0 unmount
```

Tape Subcommands

The following tape subcommands are available for the tape devices:

- **allow**

This subcommand issues the SCSI Prevent Allow Medium Removal command to the device to allow medium removal by an operator. It is normally used after the *prevent* subcommand to restore the device to the default state.

- **append**

This subcommand opens the device in *append* mode and allows appending data to the end of the current tape. The subcommand can be used with a No Rewind on Close special file to set the tape position after the last file that was written. For example:

```
# Append myfile.tar to the end of tape using dd command
tapeutil -f/dev/rmt0.1 append
dd if=myfile.tar of=/dev/rmt0
```

- **autoload**

This subcommand turns on the autoload feature only for subsequent subcommands. For example:

AIX Device Driver (Atape)

```
# Backup large.tar (requires multiple tapes) using autoloading feature
tapeutil -f/dev/rmt0 autoloading write -s large.tar
```

- **bsf [Count]**

This subcommand backward spaces the filemarks. An optional *count* can be specified. The default is 1.
- **bsr [Count]**

This subcommand backward spaces the records. An optional *count* can be specified. The default is 1.
- **compress and nocompress**

These subcommands turn the compression On and Off only for subsequent subcommands.
- **density**

This subcommand issues the SCSI Report Density command for all supported media and also for the current media loaded in the drive and displays the results. If the drive is not loaded, the current media density is not reported, and a Drive Not Ready error is returned.
- **eof [Count] and weof [Count]**

These subcommands write the filemarks. An optional *count* can be specified. The default is 1.
- **erase**

This subcommand erases the tape.
- **fsf [Count]**

This subcommand forward spaces the filemarks. An optional *count* can be specified. The default is 1.
- **fsr [Count]**

This subcommand forward spaces the records. An optional *count* can be specified. The default is 1.
- **list**

This subcommand displays the content of a tape. The output lists filemarks and the size of each record found on the tape until the end of data is reached. The output generated from this subcommand can be large, depending on the amount of data on the tape, and should usually be directed to a file. For example:

```
# List tape contents to file
tapeutil -f/dev/rmt0 list > tape.list
```
- **load**

This subcommand issues the SCSI Load command to load the next tape.
- **logsense**

This subcommand issues the STIOC_LOG_SENSE *ioctl* command and displays the data. If volume logging is active, the log sense data is also saved in the log file.
- **noautoloading**

This subcommand turns off the autoloading feature only for subsequent subcommands. For example:

```
# Make sure autoloading feature is off before writing file to tape
tapeutil -f/dev/rmt0 noautoloading write -s myfile.tar
```
- **nosili**

This subcommand turns off the SILI (Suppress Incorrect Length Indication) bit in variable length SCSI Read commands for all subsequent subcommands, such as *rtest*, *rwtest*, and *read*.

- **offline, rewoffl, and unload**

These subcommands rewind and unload the tape.

- **parms and status**

These subcommands issue the STIOCQRYP *ioctl* command and display the current tape drive, media, and device driver parameters.

- **prevent**

This subcommand issues the SCSI Prevent Allow Medium Removal command to the device to prevent medium removal by an operator until the *allow* subcommand is issued or the device is reset.

- **qrypos**

This subcommand issues the STIOCQRYPOS *ioctl* command for the logical and physical tape positions and displays the data. In addition, the current tape position is saved and can be restored using a subsequent *setpos* subcommand.

For example:

```
# Append myfile.tar to the end of tape and then read back
tapeutil -f/dev/rmt0.1 append          \
                                qrypos   \
                                write -s myfile.tar \
                                setpos    \
                                read -d temp.tar
```

```
# Verify myfile.tar was written correctly
diff myfile.tar temp.tar
```

- **read -d Destination [-c Count]**

This subcommand reads a file or a specified number of records from the tape to the destination file name specified with the *-d* flag. If the optional *count* parameter is used, only the number of records specified with the *-c* flag is read unless a filemark is encountered before the number of specified records. If the *count* parameter is not used, all records up to the next filemark on tape are read. For example:

```
# Restore myfile.tar from tape
tapeutil -f/dev/rmt0 read -d myfile.tar
```

```
# Read 3 records from the tape into myfile
tapeutil -f/dev/rmt0 read -d myfile -c3
```

- **rewind and retension**

These subcommands rewind the tape.

- **rtest [-b Blocksize] [-c Count] [-r Repetition]**

This subcommand performs a read test by reading a random data pattern from the tape and verifying that it matches the written data. The *rtest* subcommand can be used after the *wtest* subcommand to verify the data.

An optional *block size*, *count*, and *repetition* can be specified with the *-b*, *-c*, and *-r* flags, respectively. If the *block size* is fixed, the *count* specifies the number of blocks to read on each repetition. If the *block size* is zero (variable), the *count* specifies the number of bytes to read on each repetition. The default is a *block size* of 10240, a *count* of 20 blocks, and a *repetition* of 1. For example:

```
# R/W test using 256 KB blocks, 5 megabytes per write, 100 times
tapeutil -f/dev/rmt0 rewind          \
                                wtest -b 262144 -c 20 -r 100 \
                                rewind          \
                                rtest -b 262144 -c 20 -r 100
```

- **rwtest [-b Blocksize] [-c Count] [-r Repetition]**

This subcommand performs a read and write test by writing a random data pattern on the tape, reading it, and verifying that it matches the written data.

AIX Device Driver (Atape)

An optional *block size*, *count*, and *repetition* can be specified with the *-b*, *-c*, and *-r* flags, respectively. If the *block size* is fixed, the *count* specifies the number of blocks to write on each *repetition*. If the *block size* is zero (variable), the *count* specifies the number of bytes to write on each *repetition*. The default is a *block size* of 10240, a *count* of 20 blocks, and a *repetition* of 1. For example:

```
# R/W test using 256 KB blocks, 5 megabytes per write, 10 times
tapeutil -f/dev/rmt0 rwtest -b 262144 -c 20 -r 10
```

- **seod**

This subcommand spaces to the end of data on the tape.

- **setpos [Blockid]**

This subcommand issues the SCSI Locate command to the device to set the tape position. If the optional blockid parameter is specified, the tape position is set to the *blockid*. Otherwise, if the *blockid* parameter is omitted, the tape position is set to the last position saved using the *qrypos* subcommand. The *blockid* can be specified in decimal or in hex, with a leading “x”. For example:

```
# Append myfile.tar to the end of tape and then read back
tapeutil -f/dev/rmt0.1 append          \
                                qrypos   \
                                write -s myfile.tar \
                                setpos    \
                                read -d temp.tar
```

```
# Verify myfile.tar was written correctly
diff myfile.tar temp.tar
```

```
# Set tape position to block 32 and leave positioned on close
tapeutil -f/dev/rmt0.1 append setpos 32
```

```
# Set tape position to block 32 and leave positioned on close
tapeutil -f/dev/rmt0.1 append setpos x20
```

- **sili**

This subcommand turns on the SILI (Suppress Incorrect Length Indication) bit in variable length SCSI Read commands for all subsequent subcommands, such as *rtest*, *rwtest*, and *read*.

- **sync**

This subcommand synchronizes or flushes the tape buffers to tape.

- **valid "Name"**

This subcommand sets the volume ID for logging. See “Volume ID for Logging” on page 46.

- **write -s Source**

This subcommand writes the source file specified with the *-s* flag on the tape. For example:

```
# Backup myfile.tar to tape
tapeutil -f/dev/rmt0 write -s myfile.tar
```

- **wtest [-b Blocksize] [-c Count] [-r Repetition]**

This subcommand performs a write test by writing a random data pattern on the tape. The *rtest* subcommand can be used after the *wtest* subcommand to verify the data that was written.

An optional *block size*, *count*, and *repetition* can be specified with the *-b*, *-c*, and *-r* flags, respectively. If the *block size* is fixed, the *count* specifies the number of blocks to write on each *repetition*. If the *block size* is zero (variable), the *count* specifies the number of bytes to write on each *repetition*. The default is a *block size* of 10240, a *count* of 20 blocks, and a *repetition* of 1. For example:


```
# R/W test using 256 KB blocks, 5 megabytes per write, 100 times
tapeutil -f/dev/rmt0 rewind \
          wtest -b 262144 -c 20 -r 100 \
          rewind \
          rtest -b 262144 -c 20 -r 100
```

Service Aid Subcommands

The following service aid subcommands are available:

- dump [Filename]

The dump subcommand reads a dump from the device. Only on the IBM 3580 Ultrium Tape Drive, a SCSI Send Diagnostic command is issued first to force a dump.

This subcommand stores the dump in the specified *Filename* or, if *Filename* is omitted, in the system */var/adm/ras* directory. The device driver stores up to three dump files in this directory. The first dump file is named *Atape.rmtx.dump1*, where *x* is the device number, for example, *rmt0*. The second and third dump files are *dump2* and *dump3*, respectively. After a third dump file is created, the next dump starts at *dump1* again and overlays the previous dump file.

- resetdrive

This subcommand issues a Send Diagnostic SCSI command to reset the device.

Note: This subcommand is supported only on the IBM 3580 Ultrium Tape Drive.

- ucode "Name"

This subcommand downloads microcode to the device. The *Name* parameter can specify a diskette drive, such as */dev/rfd0*, or a microcode file on the RS/6000 or pSeries system. For example:

```
# download microcode from diskette
tapeutil -f/dev/rmt0 ucode /dev/rfd0
```

```
# download microcode from RISC file
tapeutil -f/dev/rmt0 ucode /etc/microcode/device.ucode
```

Block Size and SCSI Transfer Size

The minimum and maximum block sizes for the tape device and the maximum SCSI transfer size can be queried using either the interactive interface of the tape utility program and selecting **Query/Set Parameters** under Tape Commands or the command-line interface issuing the *parms* or *status* subcommand.

Configuration Parameters

The configuration parameters can be queried using either the interactive interface of the tape utility program and selecting **Query/Set Parameters** under Tape Commands or the command-line interface issuing the *parms* or *status* subcommand.

The configuration parameters can be changed temporarily using the interactive interface of the tape utility program and selecting **Query/Set Parameters** under Tape Commands.

Note: The changes are effective only while the current device is open. All configuration parameters are reset to their current default values when the device is closed.

Reserve and Release Commands

The device driver reserves the device automatically on the *open* call and releases the device on the *close* call. This prevents other applications and hosts from accessing the device. However, there can be situations when the reserve should be maintained after the *close* call. For example, some backup programs such as *tar* can open and close the device multiple times.

The reservation must be retained explicitly between the *close* call and the next *open* call.

A device can be reserved and released explicitly using either the interactive interface of the tape utility program and selecting **Reserve** or **Release** under General Commands or the command-line interface, issuing the Reserve and Release subcommands. For example:

```
# Reserve device, run tar, and then release device
tapeutil -f/dev/rmt0 reserve
tar      ... /dev/rmt0 ...
tapeutil -f/dev/rmt0 release
```

After the Reserve command is used, the device driver retains the reservation until a Release command is issued even if the device is deconfigured and reconfigured.

Tape Drive Service Aids

The service aids provided with the device driver can be called using the interactive interface of the tape utility program and selecting **Tape Drive Service Aids** under General Commands or using the Service Aid Subcommands in the command-line interface of the tape utility program. See “Service Aid Subcommands” on page 45.

Note: The AIX diagnostic subsystem must be installed in order to use **Tape Drive Service Aids** from the *tapeutil* menu.

Volume ID for Logging

The volume ID used in the log entry when volume logging is active can be set using either the interactive interface of the tape utility program and selecting **Query/Set Parameters** under Tape Commands or the command-line interface, issuing the *valid* subcommand. For example:

```
# Unload tape, load next tape, and set volume id
tapeutil -f /dev/rmt0 unload mount valid "VOLSER001"
```

Chapter 10. Tape Drive Service Aids

The service aids described here are accessible through the AIX diagnostic subsystem using the AIX *diag* command, or the interactive and command-line interfaces of the *tapeutil* program installed with the device driver. See “Service Aid Subcommands” on page 45 and “Tape Drive Service Aids” on page 46.

To access the service aids using the *diag* command:

1. Enter the *diag* command.
2. Select **Service Aids** from the Diagnostic Function Selection window.
3. Select **IBM Tape Drive Service Aids** from the Service Aid Selection window.

Note: You must have *root* authority to use the *diag* command.

To access the service aids using *tapeutil*:

1. Enter the *tapeutil* command.
2. Select **Tape Drive Service Aids** under General Commands from the window.

Note: The AIX diagnostic subsystem must be installed in order to use **Tape Drive Service Aids** from *tapeutil*.

Details of Tape Service Aids

The following service aid utilities are installed with the device driver:

- Force Microcode Dump
- Read Dump
- Microcode Load
- Error Log Analysis
- Reset Drive

Force Microcode Dump

This utility forces a dump operation on the tape drive. After the dump operation is performed, the dump data can be transferred from the tape drive using the Read Dump utility. The Force Microcode Dump utility is supported on the IBM 3580 Ultrium tape drives.

To access this utility:

1. Open the Service Aids window.
2. Select **Force Microcode Dump** from the IBM Tape Drive Service Aids window, then press Enter.
3. Select the device from the IBM Tape Drive Selection window, then press Enter.

The Force Microcode Dump operation starts, and a window opens when the operation is completed.

Read Dump

This utility transfers the dump data from the device to a file, a diskette, or a tape cartridge.

To access this utility:

1. Open the Service Aids window.
2. Select **Read Dump** from the IBM Tape Drive Service Aids window, then press Enter.
3. Select the device from the IBM Tape Drive Selection window, then press Enter.
4. Enter the destination file name or device in the Prompting for Destination window. The default destination is the `/dev/rfd0` diskette drive. To transfer the dump data to a tape cartridge, enter the device name of the tape drive (for example, `/dev/rmt0`). To transfer the dump data to a file, enter the file name. Press F7 to commit.

Note: On certain terminal types, it might be necessary to press the Esc key and the number 7 key instead of F7.

The Read Dump operation starts, and a window opens when the operation is completed.

Microcode Load

This utility downloads microcode to the device from a file or a diskette (AIX format only).

Note: To download the microcode from a DOS diskette, you must first use the AIX `dosread` command to transfer the file from the DOS diskette to the AIX file. Then you can use the Microcode Load utility to download the AIX file to the tape drive.

To access this utility:

1. Open the Service Aids window.
2. Select **Microcode Load** from the IBM Tape Drive Service Aids window, then press Enter.
3. Select the device from the IBM Tape Drive Selection window, then press Enter.
4. Enter the source file name or device on the Prompting for Source File window. The default source is the `/dev/rfd0` diskette drive. To load from a file, enter the file name. Press F7 to commit.

Note: On certain terminal types, it might be necessary to press the Esc key and the number 7 key instead of F7.

The Microcode Load operation starts, and a window opens when the operation is completed.

Error Log Analysis

This utility displays and analyzes the system error log entries for a specific tape drive and can be used for problem determination. The type of error, the SCSI command, and the sense data (if applicable) are displayed for each entry in the error log (one screen at a time).

To access this utility:

1. Open the Service Aids window.
2. Select **Error Log Analysis** from the IBM Tape Drive Service Aids window, then press Enter.
3. Select the device from the IBM Tape Drive Selection window, then press Enter.
4. If entries are listed in the error log for the selected device, then the first entry is displayed. Press Enter to display the next entry.
5. After all entries are displayed, a window opens, and the operation is completed.

Reset Drive

This utility resets the tape drive. The Reset Drive utility is supported on the IBM 3580 Ultrium tape drives.

To access this utility:

1. Open the Service Aids window.
2. Select **Reset Drive** from the IBM Tape Drive Service Aids window, then press Enter.
3. Select the device from the IBM Tape Drive Selection window, then press Enter.
The Reset Drive operation starts, and the status from the operation will be displayed when it has completed.

AIX Device Driver (Atape)

Chapter 11. Performance Considerations

This chapter describes the parameters and issues that can affect the perceived performance of the tape drive. In general, AIX applications that operate at a file level to move data between disk storage devices and tape do not exploit the full capabilities of a high end tape device. The goal of this discussion is to give an overview of the data path components involved in moving data between disk storage devices and tape. The following chapter describes basic techniques and common utilities in a specific environment that can be used to understand how a device is performing. Performance issues encountered by advanced application developers are beyond the scope of this document.

- See the hardware reference for the specific device for performance specifications.
- See the application documentation for information on device-specific application configuration.
- See the operating system documentation for information on disk storage device striping and other techniques for improving file system performance.

Data Path

The simplified model in Figure 4 shows the components involved in the data path for moving data at a file level between disk storage devices and tape.

Performance analysis must be approached by determining which component of the data path impacts performance. Typically, a performance problem can be isolated by looking at one leg of the data path at a time. The goal of this analysis is to confirm that the tape data path is not impacting the performance adversely.

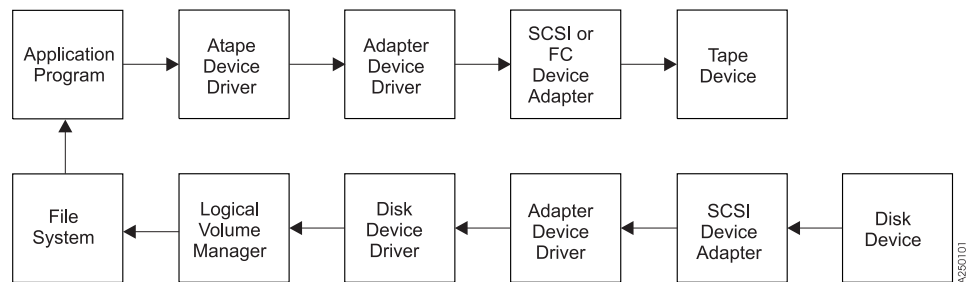


Figure 4. Data Path

Common AIX Utilities

The most commonly reported cause for poor tape performance is the use of small block sizes or the modification of the installation defaults for the tape device.

Note: The device parameters should not be changed from the defaults for most applications.

The following guidelines typically result in good tape path performance for use with AIX utilities:

1. Hardware compression should be enabled for maximum performance if the data sent to the device is uncompressed.

AIX Device Driver (Atape)

2. The *block_size* parameter should be set to variable (*block_size=0*) and command or application parameters specified to a block size appropriate for the device.
3. Block sizes of 128 KB or greater should be used to improve performance.

Before Calling Support

System performance tuning is not a support responsibility. If tests indicate raw tape performance is below specifications, record the exact failing command and collect the output from the commands in Table 5 before contacting support.

Table 5. Error Description

Information	Command
Configuration	lscfg -v
Device parameters	lsattr -E -l rmtN
Error log. Call hardware support if errors are found for TAPE_ERR* or SCSI* error labels.	errpt -a
Driver version	lslpp -l Atape.driver
Device microcode level (might not apply to all devices)	tapeutil -f /dev/rmtN reqsense
Trace of failing command	See "Trace Facility" on page 57

Chapter 12. Device and Volume Information Logging

An optional utility is provided to log information about the device and the media. This information is extensive for some devices and limited for other devices. If set to On, the logging facility gathers all available information through the SCSI Log Sense command.

This process is a separate facility from error logging. Error logging is routed to the system error log. Device information logging is sent to a separate file.

The following parameters control this utility:

- Logging
- Maximum size of the log file
- Volume ID for logging

See Chapter 3, “Tape Drive, Media, and Device Driver Parameters,” on page 15 for a description of these parameters.

Each time the rewind and unload sequence occurs or the `STIOC_LOG_SENSE ioctl` command is issued, an entry is added to the log. Each time a new cartridge is loaded, the values in the device log buffers are reset with the Log Sense command. The log data is gathered on a per-volume basis.

Log File

The data is logged in the `/usr/adm/ras` directory. The file name is dependent on each device; therefore each device has a separate log. An example of the `rmt1` device file is:

```
/usr/adm/ras/Atape.rmt1.log
```

The files are in binary format. Each entry has a header followed by the raw Log Sense pages as defined for a particular device.

The first log page is always page 0x00. This page, as defined in the SCSI-2 ANSI specification, contains all the pages supported by the device. Page 0x00 is followed by all pages specified in page 0x00. The format of each following page is defined in the SCSI specification and the device manual.

Tape Log Utility

A tape log utility is installed with the `tapelog` device driver that displays the contents of the log file in ASCII text. The log pages are displayed as hexadecimal values in dump format.

The C source code (`tapelog.c`) for the program is in the `/usr/lpp/Atape/samples` directory. The program contains a sample of the interface to the log file and the structure used to read the file.

The syntax for the tape log utility is:

```
tapelog -l Name [-d] or tapelog -f File [-d]
```

AIX Device Driver (Atape)

Notes:

1. *Name* is the logical name of the device, such as *rmt0*.
2. *File* is the name of a log file, such as *Atape.rmt0.log*.
3. The *-d* parameter, if used, deletes the log file for the specified device.

The content of the log file is displayed as standard output. To save the log in a file, use the AIX redirection function. For example:

```
tape1og -l rmt0 > rmt0.log
```

Chapter 13. Problem Determination

A set of tools is provided with the device driver to determine if the device driver and the tape device are functioning correctly. The standard AIX interface is provided for problem determination.

Error Logging

The device driver provides logging to the AIX system error log for various errors. You can view the error log for specific devices using the Error Log Analysis utility provided with the tape drive service aids. See “Error Log Analysis” on page 49. The error log can also be viewed using the *smit* or the *errpt* command.

Error Log Templates

The error log templates used by the device driver follow the same format as the default AIX tape error log entries. Each error log entry is identified by an error label and contains detail data associated with the type of error. The following describes the error labels and detail data for the templates used for logging tape device, media, and SCSI adapter related errors in the AIX system error log.

Error Labels

Errors are logged with an associated error label and error ID. The error label indicates the basic type of error:

- TAPE_ERR1
Tape media error
- TAPE_ERR2
Tape hardware error
- TAPE_ERR4
SCSI Adapter detected error
- TAPE_ERR5
Unknown error
- RECOVERED_ERROR
Temporary tape hardware or media error
- TAPE_DRIVE_CLEANSING
Tape drive needs cleaning
- DEV_DUMP_RETRIEVED
Device dump retrieved

Detail Data

Detail data is logged with the associated error that identifies the cause of the error. All error log entries use the following format for detail data:

```
Detail Data
SENSE DATA
aabb xxxx ccdd eeee eeee eeee eeee ffgg hhxx ssss ssss ssss ssss ssss
ssss ssss ssss ssss ssss ....
```

where:

aa Length of the command descriptor block (CDB)

AIX Device Driver (Atape)

- bb** SCSI target address
- xx** Unused or reserved
- cc** Start of CDB, *cc* is the operation code (byte 0)
- dd** Logical unit (byte 1) in the CDB
- ee** Bytes 2 through 12 in the CDB
- ff** Status validity field. If this field is 01, then a SCSI error was reported, and byte *gg* indicates the type of error. If this field is 02, an adapter error was reported, and byte *hh* indicates the type of error.
- gg** This byte indicates the type of SCSI error that occurred:
 - 02 CHECK CONDITION - Device reported a check condition.
 - 08 BUSY STATUS - Target is busy.
 - 18 RESERVATION CONFLICT - Target is reserved by another initiator.
 - 22 COMMAND TERMINATED - Device terminated the command.
 - 28 QUEUE FULL - Device command queue is full.
- hh** This byte indicates the type of adapter error that occurred. For parallel SCSI adapters, this is the `general_card` status code as defined in `/usr/include/sys/scsi.h`:
 - 01 HOST IO BUS ERROR - Host I/O bus error during data transfer.
 - 02 SCSI BUS FAULT - SCSI bus protocol or hardware error.
 - 04 COMMAND TIMEOUT - Command timed out before completion.
 - 08 NO DEVICE RESPONSE - Target did not respond to selection phase.
 - 10 ADAPTER HARDWARE FAILURE - Adapter indicated a hardware failure.
 - 20 ADAPTER SOFTWARE FAILURE - Adapter indicated a microcode failure.
 - 40 FUSE OR TERMINAL PWR - Blown terminator fuse or bad termination.
 - 80 SCSI BUS RESET - Adapter indicated SCSI bus has been reset.

For FCP adapters, this is the `adapter_status` code as defined in `/usr/include/sys/scsi_buf.h`:

- 01 HOST IO BUS ERROR - Host I/O bus error during data transfer.
- 02 TRANSPORT FAULT - Failure in the transport layer.
- 03 COMMAND TIMEOUT - Command timed out before completion.
- 04 NO DEVICE RESPONSE - Target did not respond to attempts to select it.
- 05 ADAPTER HARDWARE FAILURE - Adapter indicated a hardware failure.
- 06 ADAPTER SOFTWARE FAILURE - Adapter indicated a microcode failure.
- 07 WW NAME CHANGE - Adapter detected a new world wide name for the device.
- 08 FUSE OR TERMINAL PWR - Blown terminator fuse or bad termination.
- 09 TRANSPORT RESET - Adapter detected an external SCSI bus reset.
- 0A TRANSPORT BUSY - The transport layer is busy.
- 0B TRANSPORT DEAD - The transport layer is currently inoperative.

ss If byte *gg* indicates a check condition, the *ss* byte is the sense data from the device. See the appropriate device reference manual for the specific format and content of these bytes.

Automatic Dump Facility for the IBM 3580 Ultrium Tape Drive

The device driver provides an automatic dump facility for the IBM 3580 Ultrium tape drive. Whenever a check condition occurs and the sense data indicates a dump is available, the device driver reads the dump from the device and stores it in the `/var/adm/ras` directory. A maximum of three dumps for each device is stored in this directory as:

```
Atape.rmtx.dump1
Atape.rmtx.dump2
Atape.rmtx.dump3
```

where *x* is the device number, for example, *rmt0*.

When the device is first configured, the dump name is set to *dump1*. If more than three dumps occur, the driver starts over at *dump1*; therefore, the last three dumps are always kept. The device driver will also create an entry in the AIX error log labeled "DEV_DUMP_RETRIEVED" when an automatic dump was either attempted and failed or was successful.

Trace Facility

The AIX trace facility is supported for the device driver. The trace event is identified with a *hookword*. The *hookword* used by the device driver is 326. The trace can be started at any time before an operation on a tape device.

Enter the following AIX command to start the trace:

```
trace -a -j 326
```

This command starts the trace in the background and collects only the trace events with the 326 hookword (*Atape* device driver).

Enter the following AIX command to stop the trace:

```
trcstop
```

This command stops the trace after the tape operations are performed.

Enter the following AIX command to view the trace:

```
trcrpt > trace.out
```

This command formats the trace output into a readable form and places it into a file for viewing.

ATRC Utility

The *atrc* trace utility is also installed with the device driver to *start*, *stop*, and *format* a device driver trace. To start the trace, enter the *atrc* command. To stop and format the trace, enter the *atrc* command again. The trace is formatted to an *atrc.out* AIX file in the current directory.

AIX Device Driver (Atape)

Part 3. Compaq Tru64 Tape and Medium Changer Device Driver

Chapter 14. Introduction and Product Requirements

This chapter describes the Compaq Tru64 support available for the IBM 3584 UltraScalable Tape Library.

Purpose

The Tru64 Native tape and medium changer device drivers (`cam_tape`, `cam_changer`) are designed to take advantage of the features provided by tape drives and medium changer devices. The goal is to give applications access to the functions required for basic tape operations (such as backup and restore) and medium changer operations (such as mount and demount the cartridges).

Product Requirements

The following hardware and software components are supported by IBM.

Hardware Requirements

The following hardware is supported by the Tru64 Native Tape and Medium Changer driver:

- One or more of the following IBM tape and medium changer device:
 - IBM 3584 UltraScalable Tape Library with Fibre Channel interface
- One or more of the following Fibre Channel Host Bus Adapter:
 - 64 bit PCI to Fibre Channel Host Bus Adapter (Compaq part # DS-KGPSA-CA) Version: 1.32a, F/W Rev: 3.81A4

Software Requirements

Tru64 5.1A Operating System with native device driver and medium changer device driver.

Setting Up the Environment

To set up IBM Tape and Medium changer devices to operate with the Tru64 operating system:

1. Confirm that Tru64 Operating System version 5.1A is properly installed.
2. Confirm that the Host Bus Adapter card and appropriate firmware levels are properly installed on a Compaq Alpha system.
3. Connect the tape and medium changer devices to the host bus adapter card following the instructions for the devices.
4. Power up the tape and medium changer devices, if not already powered up, and wait until they have initialized.
5. Login as root administrator or use the `su` command to gain superuser privileges.
6. At the command line, issue the `/sbin/hwmgr -scan scsi` command to detect tapes or medium changers that are connected to the host.
7. To verify the scanned results, issue `/sbin/hwmgr -view dev` at the command prompt.

8. Add a new entry to the `/etc/ddr.dbase` file for the particular tape or medium changer devices. An example is provided below to illustrate a working device configuration for an IBM Ultrium Tape Drive.

```
*SCSIDEVICE
#
type = tape
Name = "IBM" "ULT3580-TD1"          #Vendor ID and Product ID
#

PARAMETERS:
    TypeSubClass          = rdat
    MaxTransferSize       = 0x0ffffff #(16M-1)
    ReadyTimeSeconds      = 45
    CMD_PreventAllow      = supported
    CMD_ExtReserveRelease = supported

DENSITY:
    DensityNumber         = 0
    DensityCode           = Default

DENSITY:
    DensityNumber         = 1
    DensityCode           = Default
    Blocking              = 0x200      #block size
    Buffered               = 1         #buffered mode

DENSITY:
    DensityNumber         = 2
    DensityCode           = Default
    CompressionCode       = 0x00      #compression off
    Blocking              = 0x8000    #block size
    Buffered               = 1         #buffered mode

DENSITY:
    DensityNumber         = 3
    DensityCode           = Default
    CompressionCode       = 0x00      #compression off
    Blocking              = 0x10000   #block size
    Buffered               = 1         #buffered mode

DENSITY:
    DensityNumber         = 4,5,6,7
    DensityCode           = Default
    CompressionCode       = 0x01      #compression on
    Blocking              = 0x20000   #block size
    Buffered               = 1         #buffered mode
```

For more information on `ddr_config` utility and the `ddr.dbase` file, refer to the Compaq Tru64 documentation and man pages.

9. Issue the `/sbin/ddr_config -c /etc/ddr.dbase` command to compile and link the new entry to the driver module.
10. The native driver will create device special files based on the information given in the `ddr.dbase` file.

Chapter 15. RAS Utility Program For Compaq Tru64 System (IBMrasutil)

IBMrasutil is a RAS utility program that allows the operator to obtain device dumps from IBM tape and medium changer devices for diagnostic purposes. It is also used to update microcode on the IBM tape and medium changer devices. The following IBM device is supported:

- IBM 3584 UltraScalable Tape Library with Fibre Channel interface

Installation Procedure from the Tape Device Drivers and SMI-S Agent CD

The xxx and X.X.X strings in the filenames are version numbers. Use the actual version numbers when issuing the commands.

1. Logon to the system as root or use *su* command to gain superuser privileges.
2. Place the IBM Ultrium Open Systems Tape Device Drivers and SMI-S Agent CD-ROM in the drive.
3. Create a directory, if necessary, to be the media mount point, such as */cdrom*:

```
mkdir /cdrom
```
4. Mount the CD-ROM on */cdrom*. For example, if the CD-ROM device is located on the C partition of *cdrom0c*, enter:

```
mount -r /dev/disk/cdrom0c /cdrom
```
5. Install the RasUtil product subset:

```
setld -l /cdrom/Drivers/Tru64/RasUtil/RasUtil.x.x.x.kit
```
6. After the installation finishes, unmount the CD-ROM:

```
umount /cdrom
```
7. After installing the IBM RAS utility program, use *vi* or some other text editor to edit the *.profile* file to add the */usr/opt/RASUTIL/bin* to the search path.
8. To find more information on how to execute the IBM RAS utility program, issue:

```
IBMrasutil -h
```

Installation Procedure from the Device Driver FTP Site

If you wish to download the RasUtil kit from the FTP site, enter this FTP address:

<ftp://ftp.software.ibm.com/storage/devdrv/Tru64>

Once you have downloaded the RasUtil kit from the FTP site, untar the kit first and then run the *setld* command to install:

```
tar -xvf RasUtil.x.x.x.kit.tar
setld -l RasUtil.x.x.x.kit
```

After installing the IBM RAS utility program, use *vi* or some other text editor to edit the *.profile* file to add the */usr/opt/RASUTIL/bin* to the search path.

To find more information on how to execute the IBM RAS utility program, issue:

```
IBMrasutil -h
```

Uninstalling

The IBM RAS Utility Program can be uninstalled by using the *setld* command:

```
setld -d IBMRASxxx
```

Update Procedure

To update the RasUtil kit to a newer version, remove the old version first.

```
setld -d IBMRASxxx
```

then follow the above installation steps to install the new version.

Query Procedure

You can determine if the IBM RAS Utility Program kit is installed on the system by using the following command.

```
setld -i|grep IBMRAS
```

To display the RasUtil product kit's fileset and the located directory, enter:

```
setld -i IBMRASxxx
```

Verify Procedure

Use this command to verify the existence of the installed RasUtil program. The *setld -v* command executes any V phase processing included in the subset control program except during installation.

```
setld -v IBMRASxxx
```

Interactive Mode

The interactive mode for the RAS utility program can be invoked from the command line by using the *IBMrasutil* command. The program will prompt you to enter a device special file name. You must open a device before you can issue any RAS utility subcommands.

Command-Line Mode

The command-line mode for the RAS utility program (*IBMrasutil*) provides the same basic RAS utility commands as the interactive mode. Invoke the *IBMrasutil* command from the Tru64 command line or from within a shell script. If you enter the *IBMrasutil* command without any arguments, the interactive mode will be invoked. The syntax for the command-line mode of the RAS utility program is:

```
IBMrasutil -f Device -Option Filename
```

Note:

1. Device is the name of the device special file (for example, */dev/ntape/tape7*)
2. Filename for input or output operations
3. If only the *-f* option is issued, the program will query and display the Model Name, Serial Number and the Firmware level of the device.
4. If the *-f* option is not issued, the interactive mode will be invoked.

Command-Line Options

-D filename

Read device dump information and output it into a file specified by the filename. For example,

```
IBMrasutil -f /dev/ntape/tape7 -D DriveDump.log
```

-M filename

Load the microcode from the specified file to the device. For example,

```
IBMrasutil -f /dev/ntape/tape7 -M 2360.bin
```

Part 4. HP-UX Tape and Medium Changer Device Driver

Chapter 16. Introduction and Product Requirements

This chapter provides an overview of the IBM Tape and Medium Changer Device Driver for HP-UX, which provides support for IBM Ultrium products (see “Hardware Requirements” for specific models).

Purpose

This device driver product (ATDD) provides SCSI-3 attachment for the IBM Ultrium products to selected Hewlett-Packard platforms running HP-UX 11.0 and 11i v1 and 11i v2.

The HP-UX installed name of this device driver is Advanced Tape Device Driver (ATDD). ATDD is used throughout this section to identify this driver.

ATDD is designed specifically to take advantage of the features provided by the IBM Ultrium tape system, including full control of the random access medium changer facility (move, element information, and inventory) present in some models. The goal is to give applications access to the functions required for basic operations (such as backup and restore) as well as the advanced functions needed by full tape management systems. Whenever possible, the device driver is designed to take advantage of the IBM tape system features transparent to the application.

Product Requirements

The following hardware and software components are required and supported by the IBM Tape and Medium Changer Device Driver for HP-UX (ATDD).

ATDD Implementation

The ATDD is supported for operation in the following HP-UX platform environment:

- HP PCI Bus - HP-UX 11.00 (64-bit) - Versions of this driver are identified by levels ATDD 1.x.x.x. rp54xx (formerly L-Class), rp74xx (formerly N-Class), and rp8400 (formerly V-Class) servers are supported.
- HP PCI Bus - HP-UX 11i v1 (64-bit) - Versions of this driver are identified by levels ATDD 3.x.x.x. rp54xx (formerly L-Class), rp74xx (formerly N-Class), and rp8400 (formerly V-Class) servers are supported.
- HP PCI Bus - HP-UX 11iv2 (64-bit for Itanium 2) - Versions of this driver are identified by levels ATDD 5.5.x.x. rx1600, rx1620, rx2600, rx2620, rx4640, and rx5670 servers are supported.
- HP PCI Bus - HP-UX 11iv2 (64-bit for PA-RISC) - Versions of this driver are identified by levels ATDD 5.6.x.x. rp54xx (formerly L-Class), rp74xx (formerly N-Class), and rp8400 (formerly V-Class) servers are supported.

Hardware Requirements

All versions of the ATDD configure and operate the following tape drives and libraries unless otherwise noted:

- IBM TotalStorage Ultrium External Tape Drive 3580
- IBM TotalStorage Ultrium Tape Autoloader 3581

HP-UX Device Driver (ATDD)

- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582
- IBM TotalStorage Ultrium Scalable Tape Library 3583
- IBM TotalStorage UltraScalable Tape Library 3584

ATDD supports the following HP SCSI host bus adapters:

- HP A4800A PCI Ultra SCSI Host Bus Adapter (HVD)
 - Support for HP rp54xx, rp74xx, and rp8400 systems
- HP A5159A PCI Dual Port FWD SCSI Host Bus Adapter (HVD)
 - Support for HP rp54xx and rp74xx systems
- HP A5149A PCI to Ultra 2 SCSI Host Bus Adapter (LVD Single Port)
 - Support for HP rp54xx, rp74xx, and rp8400 systems
- HP A5150A Dual Channel PCI to Ultra 2 SCSI Host Bus Adapter (LVD Dual Port)
 - Support for HP rp54xx and rp74xx systems
- HP A6828A Ultra 160 SCSI Adapter (LVD Single Port)
 - Support for HP rp54xx, rp74xx, and rp8400 systems
 - Support for HP rx1600, rx1620, rx2600, rx2620, rx4640, rx5670
- HP A6829A Ultra 160 SCSI Adapter (LVD Dual Port)
 - Support for HP rp54xx, rp74xx, and rp8400 systems
 - Support for HP rx1600, rx1620, rx2600, rx2620, rx4640, rx5670

ATDD supports the following HP Fibre Channel adapters:

- HP A5158A Fibre Channel Mass Storage adapter
 - Support for HP rp54xx, rp74xx, and rp8400 systems
- HP A6795A Fibre Channel PCI Tachyon XL2 host bus adapter
 - Support for HP rp54xx, rp74xx, and rp8400 systems
 - Support for HP rx1600, rx1620, rx2600, rx2620, rx4640, rx5670
- HP A6826A PCI-X Fibre Channel adapter
 - Support for HP rp54xx, rp74xx, and rp8400 systems
 - Support for HP rx1600, rx1620, rx2600, rx2620, rx4640, rx5670

Table 6. IBM Tape and Medium Changer Drivers for HP-UX (ATDD) and HP-UX System

HP Device Driver (ATDD)	Architecture	Operating System Level	Machine Types	Supported Host Adapters
ATDD 1.x.x.x	HP PCI Bus System	HP-UX 11.00 (64-bit)	HP rp54xx (formerly L-Class), rp74xx (formerly N-Class), rp8400 (formerly V-Class)	HP A4800A (HVD) (rp54xx, rp74xx, rp8400) HP A5159A (HVD) (rp54xx, rp74xx) HP A5149A (LVD) (rp54xx, rp74xx, rp8400) HP A5150A (LVD) (rp54xx, rp74xx) HP A5158A (FC) (rp54xx, rp74xx, rp8400) HP A6795A (FC) (rp54xx, rp74xx, rp8400)

Table 6. IBM Tape and Medium Changer Drivers for HP-UX (ATDD) and HP-UX System (continued)

ATDD 3.x.x.x	HP PCI Bus System	HP-UX 11i v1 (64-bit)	HP rp54xx, (formerly L-Class), rp74xx (formerly N-Class), rp8400 (formerly V-Class)	HP A4800A (HVD) (rp54xx, rp74xx, rp8400) HP A5159A (HVD) (rp54xx, rp74xx) HP A5149A (LVD) (rp54xx, rp74xx, rp8400) HP A5150A (LVD) (rp54xx, rp74xx) HP A6828A (LVD) (rp54xx, rp74xx, rp8400) HP A6829A (LVD) (rp54xx, rp74xx, rp8400) HP A5158A (FC) (rp54xx, rp74xx, rp8400) HP A6795A (FC) (rp54xx, rp74xx, rp8400) HP A6826A (FC) (rp54xx, rp74xx, rp8400)
ATDD 5.5.x.x	HP PCI Bus System	HP-UX 11i v2 (Itanium 2) (64-bit)	HP rx1600, rx1620, rx2600, rx2620, rx4640, rx5670	HP A6828A (LVD) (rx1600, rx1620, rx2600, rx2620, rx4640, rx5670) HP A6829A (LVD) (rx1600, rx1620, rx2600, rx2620, rx4640, rx5670) HP A6795A (FC) (rx1600, rx1620, rx2600, rx2620, rx4640, rx5670) HP A6826A (FC) (rx1600, rx1620, rx2600, rx2620, rx4640, rx5670)
ATDD 5.6.x.x	HP PCI Bus System	HP-UX 11i v2 (PA-RISC) (64-bit)	HP rp54xx, (formerly L-Class), rp74xx (formerly N-Class), rp8400 (formerly V-Class)	HP A6828A (LVD) (rp54xx, rp74xx, rp8400) HP A6829A (LVD) (rp54xx, rp74xx, rp8400) HP A5158A (FC) (rp54xx, rp74xx, rp8400) HP A6795A (FC) (rp54xx, rp74xx, rp8400) HP A6826A (FC) (rp54xx, rp74xx, rp8400)

HP-UX Device Driver (ATDD)

Note:

1. The ATDD does not support IBM Ultrium tape devices that are attached to the HSC/GSC bus or the Precision Bus (HP-PB) architectures.
2. For attachment of IBM Ultrium devices to HP-UX rp8400 (formerly V-Class) platforms, a feedthrough SCSI terminator, which attaches to the A4800A Host Bus Adapter, is required. Attach the male end of the feedthrough SCSI terminator to the host adapter and connect the host end of the SCSI cable to the feedthrough terminator. Terminate the SCSI bus at the last IBM Ultrium device on the bus as usual.
3. The low level FC-Tape protocol standard is not supported in the HP Fibre Channel HBA firmware and users should restrict their SAN configurations. The jobs will be aborted with HP-UX Fibre Channel HBA configurations when errors are detected in the SAN. The HP A5158A and HP A6795A host adapters are not compliant with the Fibre Channel Protocol for SCSI, second version (FCP-2), dated 17 November 2000, a draft proposed by the InterNational Committee for Information Technology Standard (INCITS).
4. Using a single Fibre Channel host bus adapter (HBA) for concurrent tape and disk operations is not recommended. Tape and disk devices require incompatible HBA settings for reliable operation and optimal performance characteristics. Under stress conditions (high I/O rates for tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems have been observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and using SAN zoning to minimize contention. IBM is focused on assuring server and storage configuration interoperability. It strongly recommends that your implementation plan includes provisions for separating disk and tape workloads.
5. The HP A6826A HBAs need the Qlogic firmware level B.11.11.02 or later. If the adapter is directly connected to a 3592 without a switch, then the topology on the drive port needs to be set to L-port.

Software Requirements

This product requires the following HP-UX patches. The patches listed might be superseded. Contact Hewlett-Packard to obtain the latest patches available.

Patches for DKLM version of ATDD (3.5.x.x or 1.15.x.x)

<u>HP-UX</u>	<u>Patch</u>	<u>Patch Description</u>
11.00	PHKL_28496	s700_800 11.00 SCSI IO Subsystem Cumulative patch
11i v1	PHKL_29818	s700_800 11.11 Cumulative DLKM module patch
11i v1	PHKL_29889	s700_800 11.11 early boot,Psets,vPar,Xserver,T600 HPMC KRNG
11i v1	PHKL_29890	s700_800 11.11 vPar enablement,DKLM load panic

Patches for Fibre Channel A5158A Adapter:

<u>HP-UX</u>	<u>Patch</u>	<u>Patch Description</u>
11.00	PHSS_23996	s700_800 11.00 Tachyon TL Fibre Channel Driver Patch
11.00	PHKL_23939	s700_800 11.00 Fibre Channel Mass Storage Driver Patch
11i v1	PHKL_23626	s700_800 11.11 Fibre Channel Mass Storage Patch

Patches for LVD A6828A and A6829A Adapters:

HP-UX	Patch	Patch Description
11.00	PHSS_30509	s700_800 11.00 SCSI Ultra160 Cumulative Patch
11i v1	PHKL_30511	s700_800 11.11 SCSI Ultra160 Cumulative Patch

Correct processor Dependent Code (PDC) firmware versions are required to support booting with the A5158A host bus adapter. Contact your HP representative to obtain the correct levels.

Patches for IBM Ultrium 3583 with the Integrated Router on HP-UX 11.00 PCI Bus Systems:

HP-UX	Patch	Patch Description
Bundles patches:		
11.00	Online Diag	B.11.00.08.07 HP-UX 11.0 Support Tools Bundle
	XSWG1100	B.11.00.47 HP-UX Extension Pack, May 1999
	XSWHWCR1100	B.11.00.47 Hardware Enablement and Critical Patches (Sep. 1999)
	Y2K-1100	B.11.00.B0315 HP-UX Core OS Year 2000 Patch Bundle
Individual patches:		
11.00	PHKL_23939	s700_800 11.00 Fibre Channel Mass Storage Driver Patch
	PHNE_15537	s700_800 11.00 Fibre Channel Cumulative Patch
	PHCO_22526	s700_800 11.00 Software Distributor (SD-UX) Cumulative Patch
	PHCO_21315	s700_800 11.00 HPArray Manager/60 Cumulative Patch
	PHCO_23262	s700_800 11.00 HP AutoRAID Manager Cumulative Patch
	PHKL_24004	s700_800 11.00 SCSI I/O Cumulative Patch

The latest driver information can be found on the distribution CD at *../HPUX/atdd.Readme*.

Data Flow

Both data and commands flow between the application program and the tape subsystem through the IBM Tape and Medium Changer Device Driver for HP-UX. Figure 5 shows the relationships between the IBM Tape and Medium Changer Device Driver for HP-UX, the application program, the adapter device driver, and the IBM tape subsystem.

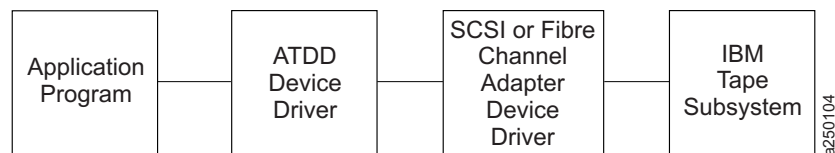


Figure 5. Data Flow

Software Interface to the Device Driver

The IBM Tape and Medium Changer Device Driver for HP-UX provides the following standard HP-UX (UNIX[®]) entry points for IBM tape subsystems:

- Open** This entry point is driven by the *open* system function call.
- Write** This entry point is driven by the *write* system function call.
- Read** This entry point is driven by the *read* system function call.
- Close** This entry point is driven explicitly by the *close* system function call and implicitly by the operating system at program termination.
- ioctl** This entry point is driven by the input/output control (*ioctl*) system function call. It provides a set of tape device, medium changer device, and SCSI specific operations. It allows HP-UX applications to access and control the features and attributes of IBM tape subsystems programmatically.

For programming information, see the *IBM Ultrium Device Drivers: Programming Reference*.

Chapter 17. Install, Uninstall, and Configure the DLKM Version of ATDD

This chapter describes how to install, manage, and uninstall the Dynamically Loadable Kernel Module (DLKM) version of IBM Tape and Medium Changer Device Driver for HP-UX (ATDD). By default, the ATDD automatically configures all supported IBM tape drives that are attached and powered On when the system starts. ACDD, the changer part of ATDD, does not configure IBM Medium Changer devices by default, because many applications use either their own changer drivers or the native *schgr* or *sctl* drivers.

During and after the ATDD is installed, ACDD, the changer part of ATDD, can be set to configure (CLAIM) all attached IBM Medium Changer devices. Additionally, selected IBM Medium Changer devices can be configured by running the *swinstall* command with the `-x ask=true` command option, or by running `/opt/atdd/bin/atdd_claim -c`. The installation process depends on whether all IBM tape drives are to be configured by ATDD or only selected ones, and whether configured tape drives exhibit default behavior or require specific configuration settings. Additionally, for IBM Medium Changers, installation selection or post-installation configuration can determine whether all IBM Medium Changers should be configured, selectively configured, or not configured at all.

Options for installation of the ATDD:

- If you are configuring all IBM tape drives with the most recent ATDD using the default settings, and configuring no medium changer devices, review the *atdd.Readme* file on the driver distribution CD and start at “Install the Driver Using the CD Installation Script” on page 78.
- If you are configuring selected IBM tape drives or need to change the operational behavior of any IBM tape device under the control of ATDD, review the *atdd.Readme* file on the driver distribution CD and start at “Create the Drive Configuration File (Optional)” on page 76.
- If you want to install a previous version of the ATDD that is available on the driver distribution CD, start at “Install Drivers Manually” on page 79.

Note the following facts about the command sequences described in this section:

- In some of the examples, filenames given on the command line must be referenced with an absolute path. Using `'pwd'/filename` to reference a file instead of *filename* ensures this.
- All SD-UX commands (for example, *swinstall* or *swcopy*) can be run first with the `-p` flag to preview the command. After observing the preview output, you can reissue the command without the `-p` flag to perform the actual operation.
- The SD-UX commands are moderately complex scripts that usually proceed in several steps. The steps are typically *Selection*, *Analysis*, and *Execution*. Each step can produce useful information and error messages. It is a good idea to carefully observe the results of the installation process as it occurs.

If you encounter unexpected results during the installation, examine the associated log file.

While using the SD-UX commands, you might encounter the following error about mounted file systems:

HP-UX Device Driver (ATDD)

ERROR: "hostname/": One or more file systems that appear in the file system table are not mounted and cannot be mounted.

Many commands (*swinstall*, *swremove*, and so on) attempt to mount all file systems in the */etc/fstab* file at the start of the analysis phase. This step ensures that all listed file systems are mounted before proceeding. This policy helps ensure that files are not loaded into a directory that could be below a future mount point, but it often causes an error with NFS-mounted file systems.

This error can be overridden using *-x* to set the *mount_all_file systems* option to *FALSE*. When this option is used, the command finishes with a warning indicating that no attempt will be made to mount all file systems. For example:

```
# swinstall -x mount_all_file systems=false -x ask=true atdd
WARNING: "hostname/": There will be no attempt to mount file systems
that appear in the file system table.
```

Create the Drive Configuration File (Optional)

If you are **not** using the standard device driver defaults, you must create a configuration file that directs the device driver on how to customize driver behavior for particular IBM Ultrium devices. If all configured devices are to use the device driver configuration defaults, it is not necessary to create a configuration file before installing the driver for ATDD.

The configuration file is named *sbin/init.d/atdd* and has the following syntax:

```
ATDD_global_parameter=value
ATDD_device_parameter[index]=value
```

- Blank lines and lines starting with # are ignored.
- No spaces are allowed within an entry.
- No trailing comments can be on a variable definition line.

Determine the Drive Hardware Path for IBM 3580 Ultrium Tape Drive, 3581 Tape Autoloader with SCSI Attachment

To determine the hardware path for the 3580 Tape Drive and the 3581 Tape Autoloader with SCSI attachment, follow this procedure:

1. Run *ioscan* to determine which SCSI adapter the IBM Ultrium drive or autoloader is connected to:

```
# ioscan -f -C ext_bus
```
2. Record the hardware path entry of the adapter, for example, 0/4/0/0.
3. Determine the SCSI target address of the IBM Ultrium drive (3580) or the drive in the IBM 3581 Ultrium Tape Autoloader. The switch on the rear of the IBM 3580 Ultrium Tape Drive displays the selected SCSI target address.

The LCD panel on the front of the unit can display the drive (and medium changer SCSI address) for the IBM 3581 Ultrium Tape Autoloader.

```
HWPATH=adapterpath.drivetargetaddress.0 (3580 drive)
```

or

```
HWPATH=adapterpath.drivetargetaddress.0 (3581 drive)
```

For an IBM 3581 Ultrium Tape Autoloader at SCSI target address 3, the drive device hardware path is: 0/4/0/0.3.0

Determine the Drive Hardware Paths for IBM Ultrium Tape Libraries with SCSI Attachment

To determine the hardware path for the 3582, 3583, and 3584 Tape Libraries with SCSI attachment, follow this procedure:

1. Run *ioscan* to determine which SCSI adapter the IBM Ultrium Tape Library is connected to:

```
# ioscan -f -C ext_bus
```

2. Record the hardware path entry of the adapter, for example:

```
1/8/0/0
```

3. Determine the SCSI target addresses of the IBM Ultrium drives in the Tape Library. The LCD panel on the front of the unit can display the drive (and medium changer) SCSI addresses.

```
HWPATH=adapterpath.drivetargetaddress.0 (drive 1)
```

```
HWPATH=adapterpath.drivetargetaddress.0 (drive n)
```

The drive hardware paths for an IBM Ultrium 3583 Tape Library with a drive at SCSI ID 3 and SCSI ID 5 are:

```
1/8/0/0.3.0
```

```
1/8/0/0.5.0
```

Determine the Drive Hardware Paths for IBM 3580 Ultrium Tape Drive, IBM Ultrium Tape Libraries with Fibre Channel Attachment

To determine the hardware path for the 3580 tape drive and the 3582, 3583, and 3584 tape library with fibre channel attachment, follow this procedure:

1. Run *ioscan* to determine which hardware path the IBM Ultrium drive is connected to:

```
# ioscan -f -C tape
```

2. Record the hardware path entry of the IBM Ultrium Fibre Channel drive. For example,

```
0/5/0/0.8.0.255.7.9.0 (3580 stand alone tape drive)
```

```
0/8/0/0.8.0.0.0.0.2 (3583 drive 1)
```

```
0/8/0/0.8.0.0.0.0.3 (3583 drive 2)
```

Create the Hardware Path Entry

If devices are to be configured with settings other than the defaults, entries defining the hardware device paths must be placed in the */etc/rc.config.d/atdd* configuration file. Create an entry for each device that requires further configuration settings. The format for the entries is

HW_PATH[index]=DeviceHardwarePath. The *index* is used to identify the device for control of the configuration settings in the next section.

For example:

```
ATDD_HWPATH[0]=0/4/0/0.1.0 (3580 stand alone SCSI drive)
```

```
ATDD_HWPATH[1]=1/10/0/0.5.0 (3581 SCSI drive 1)
```

```
ATDD_HWPATH[2]=1/10/0/0.6.0 (3581 SCSI drive 2)
```

```
ATDD_HWPATH[3]=0/5/0/0.0.0.255.7.9.0 (3580 stand alone FC drive)
```

```
ATDD_HWPATH[4]=0/8/0/0.8.0.0.0.0.2 (3583 SCSI drive 1 with SDG)
```

```
ATDD_HWPATH[5]=0/8/0/0.8.0.0.0.0.3 (3583 SCSI drive 2 with SDG)
```

HP-UX Device Driver (ATDD)

This example shows that four devices are CLAIMED by the ATDD. The first device, 3580 Stand Alone SCSI Tape Drive at SCSI target address 1, LUN 0, has *index=0*. The second device, 3581 with two 3580 SCSI Tape Drives inside at SCSI target addresses 5 and 6, LUN 0, has *index=1 and 2*. The third device, 3580 Stand Alone FC Tape Drive at target 9, LUN 0, has *index=3*. The last one is 3583 with SAN Data Gateway (SDG) Module and two 3580 SCSI Tape Drives at target 0, LUN 2 and 3 (*index=4 and 5*).

Create the Device Specific Configuration Entries (Optional)

The ATDD is shipped with default settings for all configuration parameters. If it is desired to alter these settings, an entry can be made in the configuration file, assigning an appropriate value to the desired configuration variable. Each drive can have a different value for each configuration variable. The index number associated with the configuration variable associates the parameter setting with the device at the hardware path with the same index.

Example 1:
ATDD_IMMEDIATE[0]=1

This allows application control back from the device before rewind is complete for the device at hardware path 0/4/0/0.1.0 (based on the previous hardware path entry in “Create the Hardware Path Entry” on page 77).

Example 2:
ATDD_TRAILER[1]=1

This allows *writes* after an early end-of-tape warning for the device at hardware path 0/4/0/0.1.0. See Table 8 on page 105 and “Driver-Specific Parameters” on page 106 for a description of all configuration parameters.

Note: If you are experiencing difficulties with your tape device, be sure to examine the */etc/rc.log* for errors and correct the problems.

Power Off the Tape Drives

When the ATDD software is initially installed, it attaches itself to all tape drives that are in the CLAIMED state as shown by entering the command:

```
# ioscan -fk -C tape
```

Before you continue, ensure that all devices that report CLAIMED with this command are devices you want to have managed by this device driver.

To get a tape drive out of the CLAIMED state so it will not be configured by this driver, power Off the tape drive and run *ioscan* without the *'-k'* argument as follows:

```
# ioscan -f -C tape
```

Install the Driver Using the CD Installation Script

An installation script, (*install_atdd*), is provided to automate the driver installation and perform some checking functions. It copies the latest version of the driver to the software depot and installs the driver version.

To install the ATDD with the script, insert the CD in the target system, mount the distribution CD, examine the README file, then run the install script. For example:

```
# mount -o ro /dev/cdromdevicename /cdrom
# more /cdrom/Drivers/HPUX/atdd.Readme
# /cdrom/Drivers/HPUX/install_atdd
```

When the installation script runs, the customer is prompted with the question, "Would you like to activate the Medium Changer support by ATDD(Y/N)?" Enter Y to allow the ATDD to configure all attached IBM Medium Changer devices, or answer N to allow the HP native changer driver to configure these Medium Changer devices.

Notes:

1. If a previous version of the ATDD is installed on your system, uninstall it before attempting to install the latest version. See "Uninstalling the software" on page 83.
2. If the directory `/cdrom` does not exist, create this directory using the `mkdir` command before issuing the `mount` command.
3. Typically, the special file for a CD-ROM drive has the form `/dev/dsk/cxydz` (for example: `/dev/dsk/c1t2d0`). The special file name might be different on your system.

To install the ATDD manually, use the procedures in "Install Drivers Manually."

Install Drivers Manually

Installing the drivers manually requires three steps, detailed in the following sections.

1. "Copy the Software to the Software Depot."
2. "Review the ATDD README File" on page 80.
3. "Install ATDD" on page 81.

If a previous version of the ATDD is installed on your system, uninstall it before attempting to install the latest version. See "Uninstalling the software" on page 83.

Copy the Software to the Software Depot

Attention: If you do not copy the ATDD software into a depot, you will not be able to readily uninstall the product.

If you are installing from a CD-ROM, mount the CD, examine the README file, then copy the appropriate driver to the Software Depot. For example:

```
# mount -o ro /dev/cdromdevicename /cdrom
# more /cdrom/Drivers/HPUX/atdd.Readme
# swcopy -p -s /cdrom/Drivers/HPUX/atdd.x.x.x.x atdd (preview option)
# swcopy -s /cdrom/Drivers/HPUX/atdd.x.x.x.x atdd
```

Notes:

1. If the directory `/cdrom` does not exist, create the directory using the `mkdir` command.

HP-UX Device Driver (ATDD)

- Typically, the special file for a CD-ROM drive has the form `/dev/dsk/cxydz` (for example: `/dev/dsk/c1t2d0`). The special file name might be different on your system.
- Unmount the CD-ROM before ejecting the CD. To unmount the CD-ROM, type:
/usr/sbin/umount /cdrom

If you are installing from an IBM diskette, enter:

```
# swcopy -p -s /dev/floppydevicename atdd (preview option)
# swcopy -s /dev/floppydevicename atdd
```

Note: Typically, the special file for a diskette drive has the form `/dev/floppy/cxydz`. The special file name might be different on your system.

If you do not have a diskette drive or a CD-ROM on your system, you must do the following:

- Mount the CD-ROM on another system and copy the appropriate `atdd.x.x.x.x` file in the HP-UX directory to a disk file.

For example:

```
# mount /dev/cdromdevicename /cdrom
# cp /cdrom/Drivers/HPUX/atdd.x.x.x.x atdd.depot
```

or

Use the `dd` command on another system to copy the diskette to a disk file. For example:

```
# dd if=/dev/floppy of=atdd.depot bs=1k
```

- Make the disk file accessible on the install system and use the `swcopy` command:

```
# swcopy -p -s 'pwd'/atdd.depot atdd (preview option)
# swcopy -s 'pwd'/atdd.depot atdd
```

Note: You must unmount the CD-ROM before ejecting the CD. To unmount the CD-ROM, type:

```
# /usr/sbin/umount /cdrom
```

Use `swlist` to verify that the ATDD software is in the depot:

```
# swlist -d atdd
```

Review the ATDD README File

After copying the ATDD software to the depot, use the `swlist` command to view the README file:

```
# swlist -d -a readme atdd
```

The README file lists the system configuration requirements, including required system software patches and required tape related firmware versions. The file also documents any changes in the installation, use, and administration of the software that occurred after this documentation was completed. It is therefore very important that you review it before proceeding with the software installation.

Install ATDD

When the software is in the depot and only the appropriate drives are powered On, the ATDD software can be installed to the root file system using the HP-UX *swinstall* command. If the target root file system is the primary, the ATDD software is configured automatically upon installation.

If the target is an alternate root file system, the ATDD software is not configured automatically.

Note: If an earlier version of the product is already installed on the target root file system, the existing version is replaced. This is true even if the version already installed is more recent than the version being installed.

The following commands install ATDD from the depot to the default root file system by issuing the *swinstall* command with the *ask* and *autoreboot* options set as follows:

```
# swinstall -p -x ask=true atdd (preview option)
# swinstall -x ask=true atdd
```

Running the *swinstall* command with the *ask* option set to TRUE will ask the customer, "Would you like to activate the Medium Changer support by ATDD(Y/N)?" Enter Y to allow the ATDD to configure all attached IBM Medium Changer devices, or answer N to allow the HP native changer driver to configure these Medium Changer devices.

You can use *swlist* to list the software installed on the default root file system as follows:

```
# swlist atdd
```

You can verify correct installation to the default root file system with the *swverify* command:

```
# swverify atdd
```

After the installation is completed, run the *ioscan* command to list the tape drives found:

```
# ioscan -fknC tape
Class I H/W Path Driver S/W State H/W Type Description
=====
tape 0 0/3/0/0.5.0 atdd CLAIMED DEVICE IBM 03590E11
/dev/rmt/0m /dev/rmt/0mn /dev/rmt/c4t5d0BEST /dev/rmt/c4t5d0BESTn
/dev/rmt/0mb /dev/rmt/0mnb /dev/rmt/c4t5d0BESTb /dev/rmt/c4t5d0BESTnb
tape 1 0/3/0/0.6.0 atdd CLAIMED DEVICE IBM 03590B11
/dev/rmt/1m /dev/rmt/1mn /dev/rmt/c4t6d0BEST /dev/rmt/c4t6d0BESTn
/dev/rmt/1mb /dev/rmt/1mnb /dev/rmt/c4t6d0BESTb /dev/rmt/c4t6d0BESTnb
tape 4 0/5/0/0.1.23.232.0.0.2 atdd CLAIMED DEVICE IBM ULT3580-TD1
/dev/rmt/4m /dev/rmt/4mn /dev/rmt/c6t0d2BEST /dev/rmt/c6t0d2BESTn
/dev/rmt/4mb /dev/rmt/4mnb /dev/rmt/c6t0d2BESTb /dev/rmt/c6t0d2BESTnb
tape 2 0/6/0/0.1.16.255.14.8.1 stape CLAIMED DEVICE STK 9840
/dev/rmt/c5t8d1BEST /dev/rmt/c5t8d1BESTn /dev/rmt/c5t8d1D3480 /dev/rmt/c5t8d1D3480n
/dev/rmt/c5t8d1BESTb /dev/rmt/c5t8d1BESTnb /dev/rmt/c5t8d1D3480b /dev/rmt/c5t8d1D3480nb
```

If you have enabled ACDD, run the following *ioscan* command to view the medium changers found:

HP-UX Device Driver (ATDD)

```
# ioscan -fnkC autoch
Class I H/W Path Driver S/W State H/W Type Description
=====
autoch 0 0/3/0/0.5.1 acdd CLAIMED DEVICE IBM 03590E11
/dev/rmt/0chngr
autoch 1 0/6/0/0.1.16.255.14.8.0 schgr CLAIMED DEVICE STK 9710
/dev/rac/c5t8d0
autoch 2 0/6/0/0.1.17.255.5.10.1 acdd CLAIMED DEVICE IBM 03590E11
/dev/rmt/2chngr
autoch 4 0/5/0/0.1.23.232.0.0.1 acdd CLAIMED DEVICE IBM ULT3583-TL
/dev/rmt/4chngr
```

Configure all IBM Tape Drives/Media Changers with ATDD/ACDD

By default, ATDD automatically configures all supported IBM tape drives that are attached and powered On when the ATDD package is installed. The ACDD driver does not configure IBM Medium Changer devices by default. To configure all IBM Medium Changer devices on the system during the install of ATDD package, run the *swinstall* command with the *-x ask=true* command option and answer "Yes" to allow the ACDD to claim these devices.

To configure all IBM Medium Changer devices on the system after the install of ATDD package, run the command:

```
# /opt/atdd/bin/atdd_claim -c
```

To remove the claim with ACDD after the next reboot.

```
# /opt/atdd/bin/atdd_claim -u
```

Configure Selected IBM Tape Devices with ATDD/ACDD

Attaching Selected IBM Tape Devices to ATDD/ACDD During Auto Booting

To claim a specific tape device with atdd/acdd during reboot, edit the "Claiming Device" section in the file of /sbin/init.d/atdd.

For example, to attach IBM 3590E11 medium changer with acdd driver:

1. To determine the HW path:

```
# ioscan -fnk|grep IBM
autoch 0 0/3/0/0.5.1 schgr CLAIMED DEVICE IBM 03590E11
```

2. Comment out the command line(s) in the init script of /sbin/init.d/atdd:

- For tape drives:
/opt/atdd/bin/atdd_claim -t
- For medium changers:
/opt/atdd/bin/atdd_claim -c

3. Add a command line in the "Configuring Device" section in /sbin/init.d/atdd:

```
/opt/atdd/bin/atdd_claim -H 0/3/0/0.5.1
```

Attaching Selected IBM Tape Devices to ATDD/ACDD Dynamically

To dynamically attach selected IBM tape device after the ATDD installation, run the following commands.

1. To determine the HW path of the device:

```
# ioscan -fnk|grep IBM
autoch 0 0/3/0/0.5.1 schgr CLAIMED DEVICE IBM 03590E11
```

- To claim the device with ATDD/ACDD:

```
# /opt/atdd/bin/atdd_claim -H 0/3/0/0.5.1
```

Note: The dynamic attachment could be lost after the system reboots.

Attaching Selected IBM Tape Device with a Particular Driver

Use the system *ioscan* and *rmsf* commands to force a non-atdd/acdd driver to bind an IBM tape device. For example, force bind the IBM 3590E media changer with the HP native changer driver *schgr* with these steps:

- Determine the HW path of the device:

```
# ioscan -fnk |grep IBM
tape      0 0/3/0/0.5.0  atdd      CLAIMED   DEVICE    IBM      03590E11
autoch    0 0/3/0/0.5.1  acdd      CLAIMED   DEVICE    IBM      03590E11
```

- Remove the I/O configuration from the system:

```
# rmsf -H 0/3/0/0.5.1
```

- Force binding:

```
# ioscan -M schgr -H 0/3/0/0.5.1
```

- Create the device special files:

```
# insf -H 0/3/0/0.5.1
```

- Verify the force binding:

```
# ioscan -fnk |grep IBM
tape      0 0/3/0/0.5.0  atdd      CLAIMED   DEVICE    IBM      03590E11
autoch    0 0/3/0/0.5.1  schgr     CLAIMED   DEVICE    IBM      03590E11
```

Note: The force binding will be lost after the system reboots. To keep the binding to a selected device with a particular driver after a reboot, add these command-lines in steps 2–4 in "Claiming Device" section in `/sbin/init.d/atdd` file. For example,

```
# Configuring Device Section:
# add specific device claim here, for example
# /opt/atdd/bin/atdd_claim -H 0/3/0/0.5.1
# /opt/atdd/bin/atdd_claim -H 0/6/0/0.1.30.255.0.0.1
rmsf -H 0/3/0/0.5.1
ioscan -M schgr -H 0/3/0/0.5.1
insf -H 0/3/0/0.5.1
```

Uninstalling the software

Attention: Do not try to uninstall the ATDD software by simply deleting the files that make up the ATDD fileset. This causes a number of inconsistencies on the system. It is best to use the *swremove* command.

To uninstall ATDD (from the root file system):

```
# swremove atdd
```

Other Administrative Tasks

To determine whether the current version of ATDD is installed or configured on the default root file system:

```
# swlist -a state atdd
```

To determine the versions of ATDD stored in the default depot:

```
# swlist -d atdd
```

To remove ATDD from the depot:

```
# swremove -d atdd
```

HP-UX Device Driver (ATDD)

If more than one level of ATDD exists in the depot, explicitly specify the level to remove it. For example:

```
# swremove -d atdd,r=3.5.0.0
```

To view the fileset of ATDD installed on the system:

```
# swlist -l file atdd
```

To print the status of atdd and acdd loadable modules:

```
# kadmin -Q atdd  
# kadmin -Q acdd
```

To display the values of the configuration and loadable flags:

```
# kmsystem -q atdd  
# kmsystem -q acdd
```

To query the value of the atdd and acdd tuneables:

```
# kmtune | grep atdd  
# kmtune | grep acdd
```

To load the atdd and acdd module into the running kernel:

```
# kadmin -L atdd  
# kadmin -L acdd
```

Chapter 18. Install, Uninstall, and Configure the Static Version of ATDD

This chapter describes how to install, manage, and uninstall the static version of the IBM Tape and Medium Changer Device Driver for HP-UX (ATDD). By default, the ATDD automatically configures all supported IBM tape drives that are attached and powered On when the system starts. The driver does not configure IBM Medium Changer devices by default, because many applications use either their own changer drivers or the native *schgr* or *sctl* drivers.

During and after the ATDD is installed, a kernel configuration parameter (*atdd_autoch*) can be set to allow the ATDD to configure (CLAIM) all attached IBM Medium Changer devices. Additionally, selected IBM Medium Changer devices can be configured by running the *swinstall* command with the **-x ask=true** command option, or by modifying a system header file and rebuilding the HP-UX kernel. The installation process depends on whether all IBM tape drives are to be configured by ATDD or only selected ones, and whether configured tape drives exhibit default behavior or require specific configuration settings. Additionally, for IBM Medium Changers, installation selection or post-installation configuration can determine whether all IBM Medium Changers should be configured, selectively configured, or not configured at all.

Options for installation of the ATDD:

- If you are configuring all IBM tape drives with the most recent ATDD using the default settings, and configuring no medium changer devices, review the *atdd.Readme* file on the driver distribution CD and start at “Install Drivers Manually” on page 88.
- If you are configuring selected IBM tape drives or need to change the operational behavior of any IBM tape device under the control of ATDD, review the *atdd.Readme* file on the driver distribution CD and start at “Create the Drive Configuration File (Optional)” on page 86.
- If you want to install a previous version of the ATDD that is available on the driver distribution CD, start at “Install Drivers Manually” on page 88.

Note the following facts about the command sequences described in this section:

- In some of the examples, filenames given on the command line must be referenced with an absolute path. Using *'pwd'/filename* to reference a file instead of *filename* ensures this.
- All SD-UX commands (for example, *swinstall* or *swcopy*) can be run first with the **-p** flag to preview the command. After observing the preview output, you can reissue the command without the **-p** flag to perform the actual operation.
- The SD-UX commands are moderately complex scripts that usually proceed in several steps. The steps are typically *Selection*, *Analysis*, and *Execution*. Each step can produce useful information and error messages. It is a good idea to carefully observe the results of the installation process as it occurs.

If you encounter unexpected results during the installation, examine the associated log file.

While using the SD-UX commands, you might encounter the following error about mounted file systems:

HP-UX Device Driver (ATDD)

ERROR: "hostname/": One or more file systems that appear in the file system table are not mounted and cannot be mounted.

Many commands (*swinstall*, *swremove*, and so on) attempt to mount all file systems in the */etc/fstab* file at the start of the analysis phase. This step ensures that all listed file systems are mounted before proceeding. This policy helps ensure that files are not loaded into a directory that could be below a future mount point, but it often causes an error with NFS-mounted file systems.

This error can be overridden using *-x* to set the *mount_all_file systems* option to *FALSE*. When this option is used, the command finishes with a warning indicating that no attempt will be made to mount all file systems. For example:

```
# swinstall -x mount_all_file systems=false -x ask=true -x autoreboot=true atdd
WARNING: "hostname/": There will be no attempt to mount file systems
that appear in the file system table.
```

Create the Drive Configuration File (Optional)

If you are **not** using the standard device driver defaults, you must create a configuration file that directs the device driver on how to customize driver behavior for particular IBM Ultrium devices. If all configured devices are to use the device driver configuration defaults, it is not necessary to create a configuration file before installing the driver for ATDD levels 1.7.1.0 and later.

The configuration file is named *etc/rc.config.d/atdd.cfg* and has the following syntax:

```
ATDD_global_parameter=value
ATDD_device_parameter[index]=value
```

- Blank lines and lines starting with # are ignored.
- No spaces are allowed within an entry.
- No trailing comments can be on a variable definition line.

Determine the Drive Hardware Path for IBM 3580 Ultrium Tape Drive, 3581 Tape Autoloader with SCSI Attachment

To determine the hardware path for the 3580 Tape Drive and the 3581 Tape Autoloader with SCSI attachment, follow this procedure:

1. Run *ioscan* to determine which SCSI adapter the IBM Ultrium drive or autoloader is connected to:

```
# ioscan -f -C ext_bus
```
2. Record the hardware path entry of the adapter, for example, 0/4/0/0.
3. Determine the SCSI target address of the IBM Ultrium drive (3580) or the drive in the IBM 3581 Ultrium Tape Autoloader. The switch on the rear of the IBM 3580 Ultrium Tape Drive displays the selected SCSI target address.

The LCD panel on the front of the unit can display the drive (and medium changer SCSI address) for the IBM 3581 Ultrium Tape Autoloader.

HWPATH=adapterpath.drivetargetaddress.0 (3580 drive)

or

HWPATH=adapterpath.drivetargetaddress.0 (3581 drive)

For an IBM 3581 Ultrium Tape Autoloader at SCSI target address 3, the drive device hardware path is: 0/4/0/0.3.0

Determine the Drive Hardware Paths for IBM Ultrium Tape Libraries with SCSI Attachment

To determine the hardware path for the 3582, 3583, and 3584 Tape Libraries with SCSI attachment, follow this procedure:

1. Run *ioscan* to determine which SCSI adapter the IBM Ultrium Tape Library is connected to:

```
# ioscan -f -C ext_bus
```

2. Record the hardware path entry of the adapter, for example:

```
1/8/0/0
```

3. Determine the SCSI target addresses of the IBM Ultrium drives in the Tape Library. The LCD panel on the front of the unit can display the drive (and medium changer) SCSI addresses.

```
HWPATH=adapterpath.drivetargetaddress.0 (drive 1)
```

```
HWPATH=adapterpath.drivetargetaddress.0 (drive n)
```

The drive hardware paths for an IBM Ultrium 3583 Tape Library with a drive at SCSI ID 3 and SCSI ID 5 are:

```
1/8/0/0.3.0
```

```
1/8/0/0.5.0
```

Determine the Drive Hardware Paths for IBM 3580 Ultrium Tape Drive, IBM Ultrium Tape Libraries with Fibre Channel Attachment

To determine the hardware path for the 3580 tape drive and the 3582, 3583, and 3584 tape library with fibre channel attachment, follow this procedure:

1. Run *ioscan* to determine which hardware path the IBM Ultrium drive is connected to:

```
# ioscan -f -C tape
```

2. Record the hardware path entry of the IBM Ultrium Fibre Channel drive. For example,

```
0/5/0/0.8.0.255.7.9.0 (3580 stand alone tape drive)
```

```
0/8/0/0.8.0.0.0.0.2 (3583 drive 1)
```

```
0/8/0/0.8.0.0.0.0.3 (3583 drive 2)
```

Create the Hardware Path Entry

If devices are to be configured with settings other than the defaults, entries defining the hardware device paths must be placed in the */etc/rc.config.d/atdd.cfg* configuration file. Create an entry for each device that requires further configuration settings. The format for the entries is

HW_PATH[index]=DeviceHardwarePath. The *index* is used to identify the device for control of the configuration settings in the next section.

For example:

```
ATDD_HWPATH[0]=0/4/0/0.1.0 (3580 stand alone SCSI drive)
```

```
ATDD_HWPATH[1]=1/10/0/0.5.0 (3581 SCSI drive 1)
```

```
ATDD_HWPATH[2]=1/10/0/0.6.0 (3581 SCSI drive 2)
```

```
ATDD_HWPATH[3]=0/5/0/0.0.0.255.7.9.0 (3580 stand alone FC drive)
```

```
ATDD_HWPATH[4]=0/8/0/0.8.0.0.0.0.2 (3583 SCSI drive 1 with SDG)
```

```
ATDD_HWPATH[5]=0/8/0/0.8.0.0.0.0.3 (3583 SCSI drive 2 with SDG)
```

HP-UX Device Driver (ATDD)

This example shows that four devices are CLAIMED by the ATDD. The first device, 3580 Stand Alone SCSI Tape Drive at SCSI target address 1, LUN 0, has *index=0*. The second device, 3581 with two 3580 SCSI Tape Drives inside at SCSI target addresses 5 and 6, LUN 0, has *index=1 and 2*. The third device, 3580 Stand Alone FC Tape Drive at target 9, LUN 0, has *index=3*. The last one is 3583 with SAN Data Gateway (SDG) Module and two 3580 SCSI Tape Drives at target 0, LUN 2 and 3 (*index=4 and 5*).

Create the Device Specific Configuration Entries (Optional)

The ATDD is shipped with default settings for all configuration parameters. If it is desired to alter these settings, an entry can be made in the configuration file, assigning an appropriate value to the desired configuration variable. Each drive can have a different value for each configuration variable. The index number associated with the configuration variable associates the parameter setting with the device at the hardware path with the same index.

Example 1:
ATDD_IMMEDIATE[0]=1

This allows application control back from the device before rewind is complete for the device at hardware path 0/4/0/0.1.0 (based on the previous hardware path entry in “Create the Hardware Path Entry” on page 87).

Example 2:
ATDD_TRAILER[1]=1

This allows *writes* after an early end-of-tape warning for the device at hardware path 0/4/0/0.1.0. See Table 8 on page 105 and “Driver-Specific Parameters” on page 106 for a description of all configuration parameters.

Note: If you are experiencing difficulties with your tape device, be sure to examine the */etc/rc.log* for errors and correct the problems.

Power Off the Tape Drives

When the ATDD software is initially installed, it attaches itself to all tape drives that are in the CLAIMED state as shown by entering the command:

```
# ioscan -fk -C tape
```

Before you continue, ensure that all devices that report CLAIMED with this command are devices you want to have managed by this device driver.

To get a tape drive out of the CLAIMED state so it will not be configured by this driver, power Off the tape drive and run *ioscan* without the *'-k'* argument as follows:

```
# ioscan -f -C tape
```

Install Drivers Manually

Installing the drivers manually requires three steps, detailed in the following sections.

1. “Copy the Software to the Software Depot” on page 89.
2. “Review the atdd README File” on page 90.
3. “Install ATDD” on page 90.

If a previous version of the ATDD is installed on your system, uninstall it before attempting to install the latest version. See “Uninstalling the Software” on page 94.

Copy the Software to the Software Depot

Attention: If you do not copy the ATDD software into a depot, you will not be able to readily uninstall the product.

If you are installing from a CD-ROM, mount the CD, examine the README file, then copy the appropriate driver to the Software Depot. For example:

```
# mount -o ro /dev/cdromdevicename /cdrom
# more /cdrom/Drivers/HPUX/atdd.Readme
# swcopy -p -s /cdrom/Drivers/HPUX/atdd.x.x.x.x atdd (preview option)
# swcopy -s /cdrom/Drivers/HPUX/atdd.x.x.x.x atdd
```

Notes:

1. If the directory `/cdrom` does not exist, create the directory using the `mkdir` command.
2. Typically, the special file for a CD-ROM drive has the form `/dev/dsk/cxytzyz` (for example: `/dev/dsk/c1t2d0`). The special file name might be different on your system.
3. Unmount the CD-ROM before ejecting the CD. To unmount the CD-ROM, type:
`/usr/sbin/umount /cdrom`

If you are installing from an IBM diskette, enter:

```
# swcopy -p -s /dev/floppydevicename atdd (preview option)
# swcopy -s /dev/floppydevicename atdd
```

Note: Typically, the special file for a diskette drive has the form `/dev/floppy/cxytzyz`. The special file name might be different on your system.

If you do not have a diskette drive or a CD-ROM on your system, you must do the following:

1. Mount the CD-ROM on another system and copy the appropriate `atdd.x.x.x.x` file in the HP-UX directory to a disk file.

For example:

```
# mount /dev/cdromdevicename /cdrom
# cp /cdrom/Drivers/HPUX/atdd.x.x.x.x atdd.depot
```

or

Use the `dd` command on another system to copy the diskette to a disk file. For example:

```
# dd if=/dev/floppy of=atdd.depot bs=1k
```

2. Make the disk file accessible on the install system and use the `swcopy` command:

```
# swcopy -p -s 'pwd'/atdd.depot atdd (preview option)
# swcopy -s 'pwd'/atdd.depot atdd
```

Note: You must unmount the CD-ROM before ejecting the CD. To unmount the CD-ROM, type:

HP-UX Device Driver (ATDD)

```
# /usr/sbin/umount /cdrom
```

Use *swlist* to verify that the ATDD software is in the depot:

```
# swlist -d atdd
```

Review the atdd README File

After copying the ATDD software to the depot, use the *swlist* command to view the README file:

```
# swlist -d -a readme atdd
```

The README file lists the system configuration requirements, including required system software patches and required tape related firmware versions. The file also documents any changes in the installation, use, and administration of the software that occurred after this documentation was completed. It is therefore very important that you review it before proceeding with the software installation.

Install ATDD

When the software is in the depot and only the appropriate drives are powered On, the ATDD software can be installed to the root file system using the HP-UX *swinstall* command. If the target root file system is the primary, the ATDD software is configured automatically upon installation.

Configuration requires rebooting the system and rebuilding the kernel. This requires you to issue the *swinstall* command with the *-x autoreboot* option set to TRUE, as follows.

If the target is an alternate root file system, the ATDD software is not configured automatically.

Note: If an earlier version of the product is already installed on the target root file system, the existing version is replaced. This is true even if the version already installed is more recent than the version being installed.

The following commands install ATDD from the depot to the default root file system by issuing the *swinstall* command with the *ask* and *autoreboot* options set as follows:

```
# swinstall -p -x autoreboot=true -x ask=true atdd (preview option)
# swinstall -x autoreboot=true -x ask=true atdd
```

Running the *swinstall* command with the *ask* option set to TRUE will ask the customer, "Would you like to activate the Medium Changer support by ATDD(Y/N)?" Enter Y to allow the ATDD to configure all attached IBM Medium Changer devices, or answer N to allow the HP native changer driver to configure these Medium Changer devices. Set the *autoreboot* option to TRUE and allow the system to reboot automatically during the installation of the driver.

You can use *swlist* to list the software installed on the default root file system as follows:

```
# swlist atdd
```

You can verify correct installation to the default root file system with the *swverify* command:

```
# swverify atdd
```

Post-Configuration of IBM Medium Changer Devices

After the ATDD is installed, two additional configuration choices determine whether IBM medium changers should all be configured, some selectively configured, or none configured at all. If ATDD configuration for IBM medium changers is not desired, skip this section.

ATDD supports the following IBM Ultrium medium changers:

- 3581 Ultrium Tape Autoloader
- 3581 Ultrium Tape 2U Autoloader
- 3582 Ultrium Tape Library
- 3583 Ultrium Scalable Tape Library
- 3584 UltraScalable Tape Library

There are two options to control how the ATDD handles IBM medium changers:

- Configure all IBM medium changers using kernel configuration parameter. The default for this parameter is zero, allowing *schgr* or other drives to CLAIM IBM medium changers.
- Configure selective IBM medium changers by modification of ATDD header file and kernel build.

Configure All IBM Medium Changers Using Kernel Configuration Parameter

To allow the ATDD to CLAIM all attached IBM medium changer targets, the *atdd_autoch* parameter must be enabled:

1. Call the HP System Administration Manager (SAM).
2. Select Kernel Configuration.
3. Select Configurable Parameters.
4. Scroll until you find configuration parameter *atdd_autoch* then select it (double-click).
5. A window opens - change Formula/Value to 1, then select OK or press Enter.
6. The Pending Value is now 1.
7. Exit SAM - A window opens asking to Create a New Kernel Now, defer Creation Until later, or Cancel the Modification.

When the *atdd_autoch* parameter is enabled (value=1), the ATDD CLAIMs all IBM medium changer targets that respond during the boot process. If this parameter is changed, the HP host system kernel must be rebuilt and the system rebooted. When Create a New Kernel Now is selected, the system builds a new kernel. This can take a few minutes. A second window opens that allows you to Move Kernel Into Place and Shutdown/Reboot System Now or select an option to defer the activation of the new kernel.

If you select to defer activation of the new kernel, a window opens showing the new kernel can be found in */stand/build/vmunix_test* and the configuration file used to create it can be found in */stand/build/system/SAM*. To make the kernel change effective, you must execute *usr/bin/kmupdate*, then reboot the system. The configuration file should be moved to */stand/system* at the same time.

Configure Selected IBM Medium Changers Using ATDD Header File

In addition to the preceding global option, individual IBM medium changers can be claimed by ATDD. The file `/usr/conf/space.h.d/atdd.h` (not to be confused with the user interface `/usr/include/sys/atdd.h`) can be modified by the administrator with root privileges. It is a C header file built into the kernel. To define the IBM medium changer devices to be configured by ATDD, array entries of the form `Product_ID_String:c#t#l#` must be inserted into the header file where `Product_ID_String` is the product identifier (obtained through the `ioscan` command),

- `c#` is the instance of the controller to which the changer device is attached,
- `t#` is the target ID of the device, and
- `l#` is the LUN

must be inserted in the header file.

For an IBM 3581 Ultrium Tape Autoloader:

ULT3581-TA:c#t#l#

where ULT3581 TL is the product ID, which can be obtained through `ioscan`. The description field from the `ioscan` is the combined vendor and product identifiers. The product ID portion of this string must match exactly.

c#

The instance of the controller to which the device is attached. This is not the instance of the tape driver. For example:

```
# ioscan -fknC autoch
Class  I  H/W Path      Driver S/W  State H/W Type  Description
-----
autoch 0  1/10/0/1.1.0  schgr CLAIMED  DEVICE  IBM ULT3581-TA
                               /dev/rac/c0t1d0
```

The 1/10/0/1 is the controlling instance of this device. Use `ioscan` to get the instance of the controller. For example:

```
# ioscan -fk -H 0/12/0/0
Class  I  H/W Path  Driver S/W State  H/W Type  Description
-----
ext_bus 4  0/12/0/0  c720 CLAIMED  INTERFACE  SCSI C875 Fast Wide Differential

# ioscan -fk -H1/12/0/1
Class  I  H/W Path  Driver S/W State  H/W Type  Description
-----
ext_bus 0  1/12/0/1  c720 CLAIMED  INTERFACE  SCSI C875 Fast/Wide Differential
```

a250186

This instance is 0.

t#

The target, unique ID of the device. Using the first example of `ioscan`, 1/10/0/1.1.0, the target is 1.

l#

The *lun*. From the first example, 0 is the lun.

The syntax for this example is:

```
ULT3581-TA:c0t110
```

The array looks like this:

```
char *atddBindLib[16] =
{
    "ULT3581-TA:c0t110",
    0    };
```

The contents of the *usr/conf/space.h.d/atdd.h* file are:

```
/*
 * $Header$
 */

#ifndef _H_SPACE_ATDD
#define _H_SPACE_ATDD

#include "/opt/OMImag/conf/atdd_cfg.h"

/*
 * atdd.h space definitions for atdd.
 * This file should NOT be included by user programs.
 * Before changing any value, know the ramifications of your change.
 */

#ifdef _KERNEL

/* INSERT CHANGER ARRAY ENTRIES HERE */

char *atddBindLib[16] = {
    0 /* Required - Do Not Remove */
};

int atdd_autoch          = ATDD_AUTOCH;

#endif /* _KERNEL */

#endif /* !_H_SPACE_ATDD */
```

Adding an IBM Ultrium Device Using the Currently Installed ATDD

To add an Ultrium device using the currently installed ATDD, follow this procedure:

1. Modify the */stand/system* file, adding a stanza in the form of:

```
driver 0/4/0/0.1.0    atdd
```

with the *adapter/drive* path for your device.

2. Modify */etc/rc.config.d/atdd.cfg*, adding the hardware path and instance:

```
ATDD_HWPATH[#]=0/4/0/0.1.0
```

where # denotes the next instance and the *adapter/drive* path for your device.

3. Build the kernel as root:

```
# mk_kernel -o /stand/vmunix -s /stand/system
```
4. Reboot the system:

```
# shutdown -r now
```

or

```
# reboot
```

HP-UX Device Driver (ATDD)

5. After the system is up, run `/opt/OMImag/bin/atdd_mkst` to create the new special files for the device.

```
# atdd_mkst -ti <instance>
```

where *instance* is the number from the *ioscan* output for the newly installed device. This command echoes to console but does NOT create special files. To create the special files pipe the command to *sh*.

```
# atdd_mkst -ti <instance> | sh
```

Uninstalling the Software

Attention: Do not try to uninstall the ATDD software by simply removing the files that make up the ATDD fileset. This causes a number of inconsistencies on the system. It is best to use the *swremove* command.

The SWREMOVE command with the AUTOREBOOT option must be used. This command rebuilds the kernel and removes the ATDD:

```
# swremove -p -x autoreboot=true atdd (preview option)
# swremove -x autoreboot=true atdd
```

The manual steps required include:

1. Rebuilding the kernel
2. Rebooting the system
3. Reinstalling special files for native drivers

Typically these steps are run as follows, but because the process can vary depending on details of the system installation, you are *strongly encouraged* to review the log file, which records the differences.

To rebuild the kernel, the command is:

```
# mk_kernel -o /stand/vmunix -s /stand/system
```

To reboot, use the shutdown command:

```
# shutdown -ry now
```

Other Administrative Tasks

To determine what versions of the ATDD software are currently installed on the default root file system:

```
# swlist -a state atdd
```

To determine what versions of the ATDD software are stored in the default depot:

```
# swlist -d state atdd
```

To view the set of files that are installed with the ATDD software:

```
# swlist -l file atdd
```

To remove the ATDD software from the depot:

```
# swremove -d atdd
```

If more than one level of atdd exists in the depot, explicitly specify the level to remove it. For example:

```
# swremove -d atdd,r=1.7.6.8
```

HP-UX Device Driver (ATDD)

Chapter 19. Alternate Pathing Support for Library Control Path Failover

Note: The library control path failover feature code must be installed prior to enabling the alternate pathing support in the HP-UX ATDD device driver. Refer to Table 2 on page 12 to determine which feature code is required for your machine type.

The HP-UX ATDD device driver alternate pathing support will configure multiple physical control paths to the same logical library within the device driver and provide automatic failover to an alternate control path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) that are connected to a library with two drive control ports enabled. The first HBA is connected to the first control port drive, and the second HBA is connected to the second control port drive. This connection provides two physical control paths to the same library for redundancy if one path from an HBA to the library fails.

When the host bus adapter device drivers are loaded into the running system, each HBA detects a control port to the same library, and two logical devices can be created (for example, `/dev/rmt/0chnng` and `/dev/rmt/1chnng`) by the ATDD driver or the `atdd_claim` script. Each logical device is a physical path to the same library. An application can open and use only one logical device at a time, either `0chnng` or `1chnng`, because they represent the same physical device. Without the ATDD alternate pathing support, if an application opens `0chnng` and a permanent path error occurs (because of an HBA, cable, or drive control port failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (`1chnng`), but failing transaction has to be restarted from the beginning.

When the alternate pathing support is enabled, the device driver configures them internally as a single device with multiple paths. The driver also renames the alternates paths from `1chnng` to `1chnng.altpath`. It is recommended that applications open the primary path (`0chnng`). If an application opens `0chnng` and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (`1chnng.altpath`). If successful, the current operation continues on the alternate path without interrupting the application.

Configuring and Unconfiguring Alternate Pathing Support

Alternate pathing support is not enabled automatically when the device driver is installed. The ATDD driver provides a kernel parameter `alternate_pathing` for you to enable the library control path failover. To enable the failover support in the ATDD driver, you need to perform the following steps after installing the driver:

1. Set the `alternate_pathing` parameter:

```
# /opt/atdd/bin/atdd_failover -a 1
```
2. Initiate CPF for the devices:

```
# /opt/atdd/bin/atdd_claim
```

HP-UX Device Driver (ATDD)

You can check if the driver has recognized multiple control paths for your library by running:

```
#ioscan -funC autoch
```

If you see a ".altpath" next to one of the special files, then you have successfully enabled the control path failover feature for your library.

After the alternate pathing support is enabled, it remains set until atdd_claim is ran with the alternate_pathing driver parameter set to OFF . The alternate pathing setting is retained even if the system is rebooted. If you want to turn off the control path failover feature in the ATDD driver, perform the following steps:

1. Clear the alternate_pathing parameter:

```
# /opt/atdd/bin/atdd_failover -a 0
```

2. Initiate CPF for the devices:

```
# /opt/atdd/bin/atdd_claim
```

Primary and Alternate Paths

When ATDD is loaded into the kernel, the first logical medium changer device that it sees in the system will be the primary path for that medium changer. The other logical medium changers that ATDD attached for the same medium changer will be configured as alternate paths. The device driver supports up to 16 physical paths for a single device

The primary and alternate path information can be obtained by using the following command:

```
# ioscan -funC autoch
```

The following is an example of what you might see:

```
autoch  0  0/1/0/0.101.22.255.1.1.1  acdd  CLAIMED  DEVICE  IBM  ULT3582-TL
          /dev/rmt/0chnng
autoch  1  0/1/0/0.101.22.255.1.2.1  acdd  CLAIMED  DEVICE  IBM  ULT3582-TL
          /dev/rmt/1chnng.altpath
```

Querying Primary and Alternate Path Configurations

You can display the primary and alternate path configuration for all devices by running the **ioscan** command, as explained in the previous section.

You can also display the primary and alternate paths configuration for any device by running the following command:

```
# tapeutil -f /dev/rmt/Xchnng -o phs -v
```

where Xchnng is the logical name of any device. This command shows specific information for the primary path and all alternate paths, such as the logical name of the device, the attached host bus adapter, the channel id, the target id, the logical unit number under the target, the current enabled status, and how many paths are configured for the device

Disable and Enable Primary and Alternate Paths

When you load the ATDD device driver with the `alternate_pathing` parameter to be ON, by default, all the available paths for a physical device are enabled. If, for some maintenance reason, you need to disable a path and do not want to fail over to this path, you can run the following commands, where the `-f` flag is for the primary path and the `-p` flag is for the path that you want to enable or disable.

This will disable the path:

```
# tapeutil -f /dev/Xchng -o dis -p /dev/Xchng -v
```

This will ensure that the path is labeled as "disabled."

To enable a path from a disabled state, you can run the following command:

```
# tapeutil -f /dev/Xchng -o ena -p /dev/Xchng -v
```

This will enable the path.

HP-UX Device Driver (ATDD)

Chapter 20. Special Files

For each 3580 Ultrium drive configured by the ATDD, eight special files are created. For the 3581 Autoloader, 3581 Autoloader 2U, 3582, and 3583 Library, an additional special file is created for the medium changer. Depending on the configuration of logical libraries in the 3584 tape library, additional special files are created for control ports associated with each logical library.

ATDD creates the eight tape device special files in two forms: the standard or long file name and an alternative short file name. Each set of special file names (long and short) contains four special files that have all combinations of *n* and *b* options as shown in Table 7. For more information, see the `mt(7)` man pages.

Table 7. Special Files

Special File Name	BSD Compatibility	Rewind on Close
<code>/dev/rmt/<instance#>m</code>	No	Yes
<code>/dev/rmt/<instance#>mb</code>	Yes	Yes
<code>/dev/rmt/<instance#>mn</code>	No	No
<code>/dev/rmt/<instance#>mnb</code>	Yes	No
<code>/dev/rmt/c<instance#>t<target>d<LUN>BEST</code>	No	Yes
<code>/dev/rmt/c<instance#>t<target>d<LUN>BESTb</code>	Yes	Yes
<code>/dev/rmt/c<instance#>t<target>d<LUN>BESTn</code>	No	No
<code>/dev/rmt/c<instance#>t<target>d<LUN>BESTnb</code>	Yes	No
<code>/dev/rmt/<instance#>chn</code>	N/A	N/A

To list the information about a device special file, use the `atdd_issf` program located in `/opt/OMImag/bin` for the static driver and in `/opt/atdd/bin` for the DLKM driver. For each device special_file, `atdd_issf` provides to user the HBA card instance number and HW path that the device is attached, the driver major number, the device minor number, the device file's special access modes, and the device product ID. For example,

```
# atdd_issf /dev/rmt/2mnb
atdd card instance 16 available at address 1/0/0/0.1.19.255.0.0.0 Major=51 Minor=0x1000C0 ,
settings=No Rewind ,Berkeley , Best Format IBM ULT3580-TD2

# atdd_issf /dev/rmt/6chn
atdd card instance 16 available at address 1/0/0/0.1.19.255.0.0.1 Major=51 Minor=0x100100 ,
Media Changer - IBM ULT3582-TL
```

HP-UX Device Driver (ATDD)

Chapter 21. Supported Configurations

This chapter describes the supported configuration values for multiple device types being configured on the same system. The configuration parameters are global. Because the configuration parameters are global to all devices, it is necessary to use configuration values that perform reliably on all devices attached to your system. If you are experiencing difficulties, ensure that your driver is configured properly. To check your current configuration, use the *atdd_cfg* program located in */opt/OMImag/bin* for the static driver and */opt/atdd/bin* for the DLKM driver. For program usage, do as follows:

For Static:

```
# /opt/OMImag/bin/atdd_cfg -h
```

For DLKM:

```
# /opt/atdd/bin/atdd_cfg -h
```

The configuration values for the IBM Ultrium drive are:

Device	SILI	FORCE_NARROW	DENSITY	COMPRESSION	RESERVE
IBM 3580	1	0	0	1	1

Modifying Configuration Parameters

To change a configuration parameter, use the *atdd_cfg* program located in */opt/OMImag/bin* for the static driver and */opt/atdd/bin* for the DLKM driver. You can update the current value by using the Instance Number or hardware path.

For example, to change the COMPRESSION parameter to 0 (no compression at drive) for the device at hardware path 0/4/0/0.1.0, issue the following command:

```
# atdd_cfg -g INSTANCE 0/4/0/0.1.0  
INSTANCE: 2
```

atdd_cfg returns the ATDD instance number for this device. The *instance* number is then used to set COMPRESSION to Off for this device:

```
# atdd_cfg -s COMPRESSION 0 2
```

Or, to modify COMPRESSION by using the hardware path:

```
# atdd_cfg -s COMPRESSION 0 0/4/0/0.1.0
```

To set a new BOOT default value for the configuration parameter, you **must** modify or add an entry in the *atdd.cfg* configuration file located in */etc/rc.config.d* for the static driver or the *atdd* configuration file located in */sbin/init.d*. For example, if you want the COMPRESSION default set to 0 for the device at ATDD_HWPATH[0], add the following line to the *atdd.cfg* or *atdd* file:

```
ATDD_COMPRESSION[0]=0
```

HP-UX Device Driver (ATDD)

Note: The configuration parameters are prefixed with *ATDD_* in the configuration file. For additional instructions about using the configuration program, use the *-h* option as follows:

```
# atdd_cfg -h
```

Chapter 22. Configuration Parameter Definitions

This section describes the configuration parameters and values. It is not recommended that a user modify the default settings of these parameters, except as described in Chapter 21, "Supported Configurations," on page 103. ATDD is shipped with default values that allow the most reliable execution across various device types.

Device-Specific Parameters

Some of the configuration parameters are device specific, while others are driver specific. Table 8 shows the parameters that are device specific, what they mean, and their values.

Table 8. Device-Specific Parameter Definitions

Parameter	Meaning	Values
DENSITY	Density to use when writing/reading tape	In hexadecimal
SILI	Suppress Incorrect Length Indicator	<ul style="list-style-type: none">• 0=Off (Do not suppress)• 1=On (Suppress)
BLOCKSIZE	Block Size	Size in bytes
COMPRESSION	Compression Mode	<ul style="list-style-type: none">• 0=Off (Do not use Compression at drive)• 1=On (Use Compression at drive)
BUFFERING	Buffering Mode	<ul style="list-style-type: none">• 0=Off (Do not buffer data)• 1=On (Buffer data to hardware buffers)
IMMEDIATE	Immediate Mode	<ul style="list-style-type: none">• 0=Off (Wait for rewind completion)• 1=On (Return before rewind is complete)
TRAILER	Trailer Label Processing	<ul style="list-style-type: none">• 0=Off (Do not allow writes past early end of tape (EOT) warning)• 1=On (Allow writes past early EOT warning)
ERRNO_LEOT	Error Number return for Logical End Of Tape	Value returned for writes past EOM
RESERVE	Multiple Open for the Device	<ul style="list-style-type: none">• 1=On (Reserve the device after to open)• 0=Off (Do not reserve the device after to open) (Multiple Open Support)

Driver-Specific Parameters

Some of the configuration parameters are device specific, while others are driver specific. Table 9 shows the parameters that are driver specific, what they mean, and their values.

Table 9. Driver-Specific Parameter Definitions

Parameter	Meaning	Values
INSTANCE	ATDD Device Number	(Read Only)
DEVICES	Number of Configured Devices	(Read Only)
DBG	Debug Logging	<ul style="list-style-type: none">• 0=No Debug logging• value=Mask value of desired debug level

Chapter 23. Troubleshooting

A set of tools is provided with the device driver to determine if the device driver and the tape device are functioning correctly.

Error Logging

Read the system log file (typically */var/adm/syslog/syslog.log*) and the ATDD log file (*/var/adm/atdd/atdd_d.log*) if you are having problems. The ATDD logs messages to this file, which provide information regarding the problem.

Support_info Script

Run the *support_info* script located in the */opt/OMImag/bin* directory for the static driver or */opt/atdd/bin* for the DLKM driver. This script gathers important system and configuration information. There are several sections with the keyword VERIFY, indicating information that should be verified for correctness.

Log in as user ID "root" on the system that experienced a problem. Then, run the script and redirect the output to an end of a file.

Static:

```
/opt/OMImag/bin/support_info > support_info.out
```

DLKM:

```
/opt/atdd/bin/support_info > support_info.out
```

Tracing Facility

Running the *ATDD_trace* utility located in the */opt/OMImag/bin* for the static driver or */opt/atdd/bin* for the DLKM driver displays debug information. The utility traces can be started at any time before an operation on a tape device. The default debug level is 0x00001003, which shows driver configuration and per instance drive attribute configuration and errors. You can increase the amount of debug information displayed by enabling more trace flags. Issuing *atdd_trace -f 0x1ffffff* turns on every trace.

Enter the following commands to start the trace:

1. First, determine which items need to be traced and set the trace flags using the *atdd_trace* command. For example, to trace everything:

For the static driver:

```
/opt/OMImag/bin/atdd_trace -f 0x1ffffff
```

For the DLKM driver:

```
/opt/atdd/bin/atdd_trace -f 0x1ffffff
```

The set of trace flags is displayed in the table below:

HP-UX Device Driver (ATDD)

Configuration	0x00000001
General errors	0x00000002
Routine entry points	0x00000004
Returns	0x00000008
TRACE_DBG information	0x00000010
Open/Close routine	0x00000020
Trace motion commands	0x00000040
Read/Write	0x00000100
Strategy	0x00000200
Open/Close states	0x00000400
IOCTLs	0x00000800
Trace device config routines	0x00001000
Trace scsi level stuff	0x00004000
Configuration	0x10000000
Entry point	0x01000000
General information	0x00400000
Memory	0x00200000

2. Clear the trace buffer:
For the static driver:
`/opt/OMImag/bin/atdd_trace -c`
For the DLKM driver:
`/opt/atdd/bin/atdd_trace -c`
3. To have trace data displayed immediately in a readable format as it is collected:
For the static driver:
`/opt/OMImag/bin/atdd_trace -b`
For the DLKM driver:
`/opt/atdd/bin/atdd_trace -b`
4. Start the operations on a tape device.
5. To display collected trace data in a readable format at any time:
For the static driver:
`/opt/OMImag/bin/atdd_trace -d`
For the DLKM driver:
`/opt/atdd/bin/atdd_trace -d`
6. The `atdd_trace` output can be redirected to place the trace data in a readable format into a file, by entering:
For the static driver:
`/opt/OMImag/bin/atdd_trace >trace.txt`
For the DLKM driver:
`/opt/atdd/bin/atdd_trace >trace.txt`

Atdd_d Log Daemon

The device driver provides a log daemon (*atdd_d*) facility for the IBM Ultrium Tape Drive to automatically store a dump and to log error messages into the `/var/adm/atdd` directory. The *atdd_d* log daemon is not automatically started when the driver is installed.

The following steps document how to start and configure the daemon:

1. Start the daemon by running:

For the static driver:

```
#/opt/OMImag/bin/atdd_d
```

For the DLKM driver:

```
#/opt/atdd/bin/atdd_d
```

2. Check whether the daemon is running by entering:

```
#ps -ef | grep atdd_d
```

3. To view and/or modify the settings for the daemon, enter:

For the static driver:

```
# /opt/OMImag/bin/atdd_d -h
```

For the DLKM driver:

```
#/opt/atdd/bin/atdd_d -h
```

```
usage: atdd_d [options]
  [-d <log directory>]           default is /var/adm/atdd
  [-n <maximum number of dumps>] default is 10
  [-z <maximum size of a dump>]  default is 1048576
  [-s <maximum size of a log>]   default is 100000
```

For example, to decrease the number of drive dumps to store from 10 to 6 and change the log directory to */tmp/drive_dump*, do the following:

For the static driver:

```
#/opt/OMImag/bin/atdd_d -n 6 -d/tmp/drive_dump
```

For the DLKM driver:

```
#/opt/atdd/bin/atdd_d -n 6 -d/tmp/drive_dump
```

4. View the *atdd_d.log*, which provides information about the daemon start time, message type (1 for the drive dump, 2 for SIM/MIM data, and 3 for Error message), time stamp, H/W path, and size. For example:

```
# more /var/adm/atdd/atdd_d.log ↵
Tue Dec 18 14:29:36 2001 Daemon started ↵
Tue Dec 18 14:30:36 2001 Got message type 1, created on Tue Dec 18 14:30:36 2001 ↵
Tue Dec 18 14:30:36 2001 Got dump from device 235, 0x0640c0, path 1/10/0/1.4.0,
                        size 487624 ↵
Tue Dec 18 14:30:36 2001 dump saved as /var/adm/atdd/dump.00000 ↵
Tue Dec 18 14:31:29 2001 Got message type 1, created on Tue Dec 18 14:31:29 2001 ↵
Tue Dec 18 14:31:29 2001 Got dump from device 235, 0x0640c0, path 1/10/0/1.4.0,
                        size 487624 ↵
Tue Dec 18 14:31:29 2001 dump saved as /var/adm/atdd/dump.00001 ↵
Tue Dec 18 14:33:50 2001 Got message type 1, created on ↵
Tue Dec 18 14:33:50 2001 ↵
Tue Dec 18 14:33:50 2001 Got dump from device 235, 0x1282c0, path 0/5/0/0.1.17.239.0.8.2,
                        size 186068 ↵
Tue Dec 18 14:33:50 2001 dump saved as /var/adm/atdd/dump.00002 ↵
Tue Jan 22 16:24:58 2002 Got message type 2, created on Mon Jan 21 13:18:45 2002 ↵
Tue Jan 22 16:24:58 2002 Got SIM/MIM from device 2, 0x040040, path 0/12/0/0.0.0 ↵
Tue Jan 22 16:24:58 2002 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF ↵
Tue Jan 22 16:24:58 2002 ----- ↵
Tue Jan 22 16:24:58 2002 31 00 00 44 00 00 61 40 02 30 30 30 30 30 30 30 1..D..a@.00000000 ↵
Tue Jan 22 16:24:58 2002 32 32 32 37 36 32 35 35 36 30 31 30 30 30 33 41 222762556010003A ↵
Tue Jan 22 16:24:58 2002 32 41 00 00 00 00 00 00 00 00 30 38 30 30 30 49 42 2A.....08000IB ↵
Tue Jan 22 16:24:58 2002 4D 31 33 2D 30 30 30 30 30 30 30 35 31 30 30 37 M13-000000051007 ↵
Tue Jan 22 16:24:58 2002 30 33 35 39 30 48 31 41 03590H1A00051007 ↵
Tue Jan 22 16:24:58 2002 Got message type 2, created on Mon Jan 21 14:39:00 2002 ↵
Mon Feb 4 17:06:45 2002 Daemon started ↵
Mon Feb 4 17:06:45 2002 Got message type 2, created on Mon Feb 4 17:02:57 2002 ↵
Mon Feb 4 17:06:45 2002 Got SIM/MIM from device 2, 0x064040, path 1/10/0/1.4.0 ↵
Mon Feb 4 17:06:45 2002 SIM/MIM saved as /var/adm/atdd/simmim.00000
```

AZ50188

Note: No entry is made in the log file when the daemon stops. Run `#ps -ef | grep atdd_d` to check if the daemon is stopped.

5. Stop the daemon by using the following commands:

HP-UX Device Driver (ATDD)

```
Get the daemon process id:    #ps -ef | grep atdd_d
kill the daemon process:     #kill -9 process_id
```

Problems and Solutions

Table 10 describes problems and possible solutions for errors you might encounter with the DLKM driver.

Table 10. Problems, Reasons, and Solutions

Problem	Reason and Solution
The system hangs running rmsf -H on HP-UX 11i.	DLKM Infrastructure does not set the correct flag in the device switch table entry for a DLKM driver that uses. Install HP patch PHKL_29818
ATDD DLKM modules aren't loaded in the running kernel. A message in the file of /var/adm/syslog/syslog.log "vmunix: WARNING: modpath: DLKM is not initialized"	The current kernel is not matching the symbol table (DLKM) in /stand/dlkm. Need to rebuild the kernel: <pre>#cd /stand/build #/usr/sbin/sysadm/system_prep -s /stand/build/system #/usr/sbin/mk_kernel -s /stand/build/system #mv /stand/system /stand/system.prev #mv /stand/build/system /stand/system #kupdate /stand/build/vmunix_test #shutdown -ry now</pre>
A message in the file of /var/adm/syslog/syslog.log "modld: Attempt to load unregistered module "	ATDD DLKM Module's component files aren't copied to certain subdirectories of /usr/conf and /stand. Remove ATDD and try to install it again.
# kmtune grep atdd kmtune: Cannot open file -- /stand/.kmsystune_lock	.kmsystune_lock was removed manually create a .kmsystune_lock again <pre># touch /stand/.kmsystune_lock</pre>
IBM tape devices not claimed in ioscan -fnk display	As "root", verify the drivers loaded in the running kernel using `kmadmin -s` first. If "LOADED" status, run 'atdd_claim -b' to claim all IBM devices, run 'atdd_claim -t' to claim the tape drives, run 'atdd_claim -c' to claim the changers. If "UNLOADED" run 'kmadmin -L atdd' to load the ATDD run 'kmadmin -L acdd' to load the ACDD, run 'atdd_claim' to claim the tape devices If it is still in "UNLOADED", check the entries of 'atdd' and 'acdd' in /etc/loadmods. If no atdd or acdd entries exist, the driver might be statically bound to the kernel. Or the ATDD package might not be installed. Check the installation running # <i>swlist atdd</i>
No special files found in /dev/rmt	Execute the atdd_mksf and acdd (if you have changers managed by the acdd driver) scripts found in /opt/atdd/bin atdd_mksf and acdd_mksf Note: The script prints out the commands that are necessary to create the special files. To create the files you must "pipe" the output to shell (sh). <pre>atdd_mksf sh acdd_mksf sh</pre>
The special files for a device are duplicates.	Remove all of the special files for the device in /dev/rmt Create the special files again running atdd_mksf or acdd_mksf
No special files are created by HP Stape and Schgr after installing ATDD	Run # insf -e to create the special files

Table 10. Problems, Reasons, and Solutions (continued)

Problem	Reason and Solution
Cannot open Special File and the system log has the following message: Invalid SCSI request in data at bit 7 of byte 4	Wrong DENSITY setting. Check the current value by <code>atdd_cfg -g DENSITY</code> Try setting to 0 (zero) <code>atdd_cfg -s DENSITY 0</code> See section 5.1
An attempt to read data times out and returns an error and the system log has the following messages: A SCSI command timed out and was aborted.	Make sure the SILI configuration parameter is "1". This can be checked by executing <code>atdd_cfg -g SILI</code> If the SILI = 0, try setting <code>atdd_cfg -s SILI 1</code> See section 5.1

Table 11 describes problems and possible solutions for errors you might encounter with the static driver.

Table 11. Problems and Solutions

Problem	Solution
No special files found in <code>/dev/rmt</code>	Issue the following commands: <code>cd /opt/OMImag/bin</code> <code>atdd_mkfs sh</code>
Missing special files	If you are not using the standard device driver defaults, verify that the hardware path for the device is listed in the configuration file: <code>/etc/rc.config.d/atdd.cfg</code> If the hardware path is missing or incorrect, the driver was not installed properly.
Claimed by HP Stape driver	Run <code># lsdev</code> to check that the <code>atdd</code> is in the current kernel: 1. If the <code>atdd</code> isn't in the output, rebuild the kernel or reinstall <code>atdd</code> ; 2. If the <code>atdd</code> is in the output a. rebuild the system I/O configuration: 1) Remove <code>/stand/ioconfig</code> and <code>/etc/ioconfig</code> after to backup the files; 2) Reboot the system; 3) On System Console, enter <code># /sbin/ioint -c</code> while the autobooting is halted; 4) Run <code># /sbin/insf -e</code> 5) Enter <code># ^D</code> to exit the I/O configuration. b. Consider reinstalling the HP-UX operating system if the system is new or this is a new update.
Cannot open Special File. The system log has the following message: <i>Invalid SCSI request in data at bit 7 of byte 4</i>	Wrong DENSITY setting. Check the current value by: <code>atdd_cfg -g DENSITY <atdd_inst></code> Try setting it to 0 (zero): <code>atdd_cfg -s DENSITY 0 <atdd_inst></code>
An attempt to read data times out and returns an error. The system log has the following messages: <i>A SCSI command timed out and was aborted.</i> <i>LLIO: Detected PowerFail (probably caused by SCSI bus or device reset).</i>	Check that the SILI configuration parameter is 1 by executing: <code>atdd_cfg -g SILI <atdd_inst></code> If the SILI parameter is zero, try setting it to 1: <code>atdd_cfg -s SILI 1 <atdd_inst></code>

HP-UX Device Driver (ATDD)

Table 11. Problems and Solutions (continued)

Problem	Solution
An attempt to read data times out and returns an error. The system log has the following messages: <i>A SCSI command timed out and was aborted.</i> <i>scsi3:date code...</i> <i>scsi3:timestamp...</i>	Check that the FORCE_NARROW parameter is valid for this device: <i>atdd_cfg -g FORCE_NARROW <atdd_inst></i> If FORCE_NARROW is 1, try setting it to 0: <i>atdd_cfg -s FORCE_NARROW 0 <atdd_inst></i>

Chapter 24. Tape Utility Program (tapeutil)

This chapter describes how to install, uninstall and use the IBM HP-UX Tape Utility Program, *tapeutil*.

Command Sequence Information

Take note of the following facts about the command sequences described in this section:

- In some of the examples, filenames given on the command-line must be referenced with an absolute path. Using `'pwd'/filename` to reference to a file instead of *filename* ensures this.
- All the software distributor commands (for example, *swinstall* or *swcopy*) can first be run with the `-p` flag to preview the command. After observing the preview output, reissue the command without the `-p` flag to perform the actual operation.
- The software distributor commands are moderately complex scripts that frequently require several steps. The steps are typically *Selection*, *Analysis*, and *Execution*. Each step can produce useful information and error messages, so it is a good idea to carefully observe the results of the installation process as it occurs.

If you run into unexpected results during the installation, check the associated log file.

Product Requirements

The following hardware and software components are required and supported by IBM and are necessary for implementation of the *tapeutil*.

Tapeutil Implementation

The *tapeutil* program contains utility implementations for operation with the HP PCI Bus. If you are using the ATDD version 1.7.7.2 or later, on HP-UX 11.00, use *tapeutil.hpux.4.0.0.0* or later. If you are using the ATDD version 3.0.0.1 or later, on HP-UX 11i, use *tapeutil.hpux.4.0.0.0* or later. Otherwise, use *tapeutil* level *tapeutil.hpux.3.x.x.x*.

Hardware Requirements

All versions of *tapeutil* operate the following tape devices:

- IBM TotalStorage Ultrium External Tape Drive 3580
- IBM TotalStorage Ultrium Tape Autoloader 3581
- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582
- IBM TotalStorage Ultrium Scalable Tape Library 3583
- IBM TotalStorage UltraScalable Tape Library 3584

Install tapeutil Using the CD Installation Script

An installation script, (*install_tapeutil*), is provided to automate the utility installation and perform some checking. It copies the latest version to the software depot and installs it.

To install the *Tape Utility Program* with the script, insert the CD in the CD-ROM drive, mount the distribution CD, check the README file, then run the install script. For example:

```
# mount -o ro /dev/cdromdevicename /cdrom
# more /cdrom/Drivers/HPUX/tapeutil.hpux.Readme
# /cdrom/Drivers/HPUX/install_tapeutil
```

Notes:

1. If the directory */cdrom* does not exist, create the directory using the *mkdir* command.
2. Typically, the special file for a CD-ROM drive has the form */dev/dsk/cxydz* (for example, */dev/dsk/c1t2d0*). The special file name might be different on your system.
3. Unmount the CD-ROM before ejecting the CD. To unmount the CD-ROM, type:

```
# /usr/sbin/umount /cdrom.
```

To install *tapeutil* manually, use the procedures in “Install *tapeutil* Manually” on page 114 below.

Install tapeutil Manually

Installing the product manually requires the three steps detailed in the following sections:

1. “Copy the Software to the Software Depot” on page 114
2. “Review the *tapeutil* README File” on page 115
3. “Install *tapeutil*” on page 115

Copy the Software to the Software Depot

If you are installing from a CD-ROM, mount the CD, check the README file, then copy the software to the Software Depot. For example:

```
# mount -o ro /dev/cdromdevicename /cdrom
# more /cdrom/Drivers/HPUX/tapeutil.hpux.Readme
# swcopy -p -s /cdrom/Drivers/tapeutil.hpux.x.x.x.x tapeutil (preview option)
# swcopy -s /cdrom/Drivers/tapeutil.hpux.x.x.x.x tapeutil
```

Notes:

1. If the directory */cdrom* does not exist, create the directory using the *mkdir* command.
2. Typically, the special file for a CD-ROM drive has the form */dev/dsk/cxydz* (for example, */dev/dsk/c1t2d0*). The special file name might be different on your system.
3. Unmount the CD-ROM drive before ejecting the CD. To unmount the CD-ROM drive, type

```
# /usr/sbin/umount /cdrom
```

If you do not have a CD-ROM drive on your system, you must do the following:

1. Mount the CD on another system and copy the appropriate *tapeutil.hpux.x.x.x.x* file in the HP-UX directory to a disk file.

```
# mount /dev/cdromdevicename /cdrom
# cp /cdrom/Drivers/HPUX/tapeutil.hpux.x.x.x.x tapeutil.x.x.x.x
```

or

Use the *dd* command on another system to copy the diskette to a disk file. For example:

```
# dd if=/dev/floppy of=tapeutil.depot bs=1k
```

2. Make the disk file accessible on the install system and use the *swcopy* command:

```
# swcopy -p -s 'pwd' tapeutil.depot tapeutil (preview option)
# swcopy -s 'pwd' tapeutil.depot tapeutil
```

Note: You must unmount the CD-ROM drive before ejecting the CD. To unmount the CD-ROM drive, type:

```
# /usr/sbin/umount /cdrom
```

Use *swlist* to verify that the *tapeutil* software is in the depot:

```
# swlist -d tapeutil
```

Review the *tapeutil* README File

After copying the *tapeutil* software to the depot, use the *swlist* command to view the README file in the directory */opt/tapeutil*:

```
# swlist -d -a readme tapeutil
```

The README file lists the system configuration requirements, including required system software patches and required tape related firmware versions. The file also documents any changes in the installation, use, and administration of the software that occurred after this documentation was completed. Review it before proceeding with the software installation.

Install *tapeutil*

When the software is in the depot, the *tapeutil* software can be installed to the root file system using the HP-UX *swinstall* command.

The following commands install *tapeutil* from the depot to the default root file system:

```
# swinstall -p tapeutil (preview option)
# swinstall tapeutil
```

You can use *swlist* to list the software installed on the default root file system as follows:

```
# swlist tapeutil
```

HP-UX Device Driver (ATDD)

You can verify correct installation to the default root file system with the *swverify* command:

```
# swverify tapeutil
```

Uninstalling tapeutil

Attention: Do not try to uninstall the ATDD software by simply deleting the files that make up the ATDD fileset. This causes a number of inconsistencies on the system. It is best to use the *swremove* command.

To remove the *tapeutil* software from the root file system, enter:

```
# swremove -p tapeutil (preview option)
# swremove tapeutil
```

Other Administrative Tasks

To determine the version of *tapeutil* currently installed on the default root file system:

```
# swlist -a state tapeutil
```

To determine the version of *tapeutil* stored in the default depot:

```
# swlist -d state tapeutil
```

To view the set of files installed with *tapeutil*:

```
# swlist -l file tapeutil
```

To remove *tapeutil* from the depot:

```
# swremove -d tapeutil
```

Using the tapeutil Program

A Tape and Medium Changer Utility Program called *tapeutil* is provided with the IBM Tape and Medium Changer Device Driver for HP-UX and installed in the */usr/bin* directory. The *tapeutil* program fulfills several purposes:

- The program provides the following service aids for IBM tape systems:
 - Query Device Type/Verify Device Attachment
 - Query Device Serial Number
 - Query Device Microcode Level
 - Force Device Diagnostic Dump
 - Store Device Diagnostic Dump
 - Download Device Microcode
- The program provides a menu driven test tool for exercising or testing IBM tape and medium changer devices with a full suite of supported operations:
 - Reading/Writing Data
 - Tape Motion Commands
 - Setting/Displaying Device Information/Status

- Mounting/Demounting Cartridges
- Cartridge Inventory
- In addition to the menu driven front end, the *tapeutil* program provides a command-line interface, convenient for use in shell scripts.
- The source code for the *tapeutil* program is provided for example purposes and is installed in the */opt/tapeutil* directory during the *tapeutil* package installation. This source code is commented and demonstrates calls to all the supported device driver entry points and *ioctl* commands, giving the application developer a starting point for interfacing to the HP-UX device driver.

Interactive Mode

The *tapeutil* program provides both an interactive mode and a command-line mode. If the *tapeutil* program is called with no command-line parameters, the interactive mode version is started. In the interactive mode, the device to be operated on should first be opened using option 1. Other options can then be selected. The user is prompted for additional information if required for the specific options selected. The results of a command are displayed after it is executed. If an error occurs for the command, error information and device sense data are displayed. The device can be closed using option 2, or it is closed automatically when the *Quit* option is selected. The menu is displayed once automatically when the program is first called. To prevent unnecessary scrolling of the screen, the menu is not displayed again automatically after each command but instead is refreshed only after the *M* (*menu refresh*) command is entered.

These commands can issue the *MTIOCTOP ioctl* with the MT opcodes defined in */usr/include/sys/mtio.h* (MT mode), or issue the *STIOC_TAPE_OP ioctl* with the *ST_OP* opcodes defined in */usr/include/sys/st.h* (ST mode). For detailed information, see the *MTIOCTOP* and *STIOC_TAPE_OP* sections in the *IBM Ultrium Device Drivers: Programming Reference*, GC35-0483.

The default for *tapeutil* is the ST mode. Toggle between the MT or ST mode by using option 8. The following commands run in the two modes:

- Write File Mark
- Erase Tape
- Rewind
- Offline
- Forward/Backward Space File
- Forward/Backward Space Record
- Locate End of Data

Figure 6 on page 118 and Figure 7 on page 118 show the menus that are displayed by the *tapeutil* program for HP-UX 11.00 and 11i PCI Bus Systems.

HP-UX Device Driver (ATDD)

```

+-----+
| IBM SCSI TAPE & MEDIUM CHANGER UTILITY PROGRAM |
+-----+
| << GENERAL COMMANDS >> | << SERVICE COMMANDS >> |
| 1: Open Device          | 3: Query Serial Number |
| 2: Close Device        | 4: Query Microcode Level |
| D: Device Type         | 5: Force Dump          |
| M: Menu Refresh        | 6: Store Dump          |
| Q: Quit Program        | 7: Download Microcode  |
|                         | 8: Query Driver Level  |
+-----+
| << BASIC SCSI COMMANDS >> |
| 9: Test Unit Ready     | 10: Inquiry            | 11: Request Sense |
| 12: Log Sense Page     | 13: Mode Page         | 14: Reserve on Close |
| 15: Release            | 16: Prevent/Allow Media Removal |
+-----+
| << TAPE DRIVE COMMANDS >> |
| 26: Read Data          | 27: Write Data         |
| 28: Write File Mark    | 29: Erase Tape         |
| 30: Rewind             | 31: Retension          |
| 32: Offline            | 33: Load/Unload Tape  |
| 34: Forward Space File | 35: Backward Space File |
| 36: Forward Space Record | 37: Backward Space Record |
| 38: Locate End of Data | 39: Get Parameter      |
| 40: Set Parameter      | 41: Sync Buffer         |
| 42: Display Message    | 43: Report Tape Density |
| 44: Check Device Type  | 45: Get Record Size    |
| 46: Set Record Size    | 47: Get Device Status  |
| 48: Get Device Info    | 49: Get Media Info     |
| 50: Get Position       | 51: Set Position       |
| 52: Set MT/ST Mode     |
+-----+

```

Figure 6. TAPEUTIL Program Menu for the tape drive on HP-UX 11.00, 11i v1, and 11i v2 PCI Bus Systems

```

+-----+
| IBM SCSI TAPE & MEDIUM CHANGER UTILITY PROGRAM |
+-----+
| << GENERAL COMMANDS >> | << SERVICE COMMANDS >> |
| 1: Open Device          | 3: Query Serial Number |
| 2: Close Device        | 4: Query Microcode Level |
| D: Device Type         | 5: Force Dump          |
| M: Menu Refresh        | 6: Store Dump          |
| Q: Quit Program        | 7: Download Microcode  |
|                         | 8: Query Driver Level  |
+-----+
| << BASIC SCSI COMMANDS >> |
| 9: Test Unit Ready     | 10: Inquiry            | 11: Request Sense |
| 12: Log Sense Page     | 13: Mode Page         | 14: Reserve on Close |
| 15: Release            | 16: Prevent/Allow Media Removal |
+-----+
| << MEDIUM CHANGER COMMANDS >> |
| 17: Move Medium       | 18: Exchange Medium    | 19: Position to Elem |
| 20: Element Info      | 21: Inventory          | 22: Read Device IDs  |
| 23: Audit              | 24: Audit Range        | 25: Lock/Unlock Door |
+-----+
| << CPF COMMANDS >> |
| 53: Display Paths     | 54: Enable Path        | 55: Disable Path    |
+-----+

```

Figure 7. TAPEUTIL Program Menu for the medium changer on HP-UX 11.00, 11i v1, and 11i v2 PCI Bus Systems

Command-Line Interface

If command-line parameters are provided when the program is started, the program runs in command-line mode. When in command-line mode, the device is first opened, the specific command is issued, and the device is then closed. The program can be invoked from within a shell script if desired. Results of the operation are displayed only when executed in verbose mode. No information is displayed when not in verbose mode. This is particularly useful for quiet shell script implementations. A completion code, as defined in `/usr/include/sys/errno.h`, for the operation requested, is always returned from the program upon exit (in both verbose and quiet mode)

The usage of the `tapeutil` program in command-line mode is as follows:

```
tapeutil -f device -o operation [options]
```

where **device** is the name of the tape device special file (for example, `/dev/rmt/1m`) and **operation** is one of the values listed below. The device special file and the operation are required. The specific **options** associated with a particular operation are indicated below. Parameters enclosed in square brackets are optional. All others are required.

Service Commands

Query Serial Number

```
tapeutil -f f -o qsn [-w w] [-v]
```

Query Microcode Level

```
tapeutil -f f -o qmc [-w w] [-v]
```

Force Dump

```
tapeutil -f f -o fdp [-w w] [-v]
```

Store Dump

```
tapeutil -f f -o sdp [-w w] [-v] -z z
```

Download Microcode

```
tapeutil -f f -o dmc [-w w] [-v] -z z
```

Query Device Type

```
tapeutil -f f -o chk [-w w] [-v]
```

Basic SCSI Commands

Test Unit Ready

```
tapeutil -f f -o tur [-w w] [-v]
```

Inquiry/Inquiry Page

```
tapeutil -f f -o inq [-w w] [-v] [-t t] [-x x]
```

Request Sense

```
tapeutil -f f -o req [-w w] [-v]
```

Log Sense Page

```
tapeutil -f f -o log [-w w] [-v] [-x x]
```

Mode Page

```
tapeutil -f f -o mod [-w w] [-v] -x x
```

Reserve

```
tapeutil -f f -o res [-w w] [-v]
```

Release

```
tapeutil -f f -o rel [-w w] [-v]
```

Prevent/Allow Medium Removal

```
tapeutil -f f -o rem [-w w] [-v]-x x
```

Medium Changer Commands

Move Medium

```
tapeutil -f f -o mov [-w w] [-v] -s s -d d
```

Position To Element

```
tapeutil -f f -o pos [-w w] [-v] -d d
```

Element Information

```
tapeutil -f f -o ele [-w w] [-v]
```

Inventory

```
tapeutil -f f -o inv [-w w] [-v]
```

Output inventory data into a file

```
tapeutil -f f -o inv [-w w] [-v] -t 1 > filename
```

Audit

```
tapeutil -f f -o aud [-w w] [-v]
```

Lock/Unlock Door

```
tapeutil -f f -o lck [-w w] [-v] -x x
```

Exchange Medium

```
tapeutil -f f -o exc [-w w] [-v] -s s - -d d1 d2
```

Read Element Status with Drive ID

```
tapeutil -f f -o dvc [-w w] [-v] -s s - -d d1 d2
```

Initial Element Status Range

```
tapeutil -f f -o ier [-w w] [-v] -s s - -x x
```

CPF Commands

Display Paths

```
tapeutil -f f -o phs [-v]
```

Enable Path

```
tapeutil -f f -o ena -p path [-v]
```

Disable Path

```
tapeutil -f f -o dis -p path [-v]
```

Tape Drive Commands

```
Read tapeutil -f f -o rea [-w w] [-v] -b b -n n -m m
```

```
Write tapeutil -f f -o wri [-w w] [-v] -b b -n n -m m [-r r] [-z z]
```

Write File Mark

```
tapeutil -f f -o eof [-w w] [-v] -c c
```

Erase Tape

```
tapeutil -f f -o era [-w w] [-v]
```

Rewind

```
tapeutil -f f -o rew [-w w] [-v]
```

Retention

```
tapeutil -f f -o ret [-w w] [-v]
```

Offline

```
tapeutil -f f -o off [-w w] [-v]
```

Load/Unload Tape

```
tapeutil -f f -o lod [-w w] [-v] -x x
```

Forward Space File

```
tapeutil -f f -o fsf [-w w] [-v] -c c
```

Backward Space File

```
tapeutil -f f -o bsf [-w w] [-v] -c c
```

Forward Space Record

```
tapeutil -f f -o fsr [-w w] [-v] -c c
```

Backward Space Record

```
tapeutil -f f -o bsr [-w w] [-v] -c c
```

Locate End of Data

```
tapeutil -f f -o eod [-w w] [-v]
```

Report Tape Density

```
tapeutil -f f -o den [-w w] [-v]
```

Check Device Type

```
tapeutil -f f -o chk [-w w] [-v]
```

Get Record Size

```
tapeutil -f f -o grs [-w w] [-v]
```

Set Record Size

```
tapeutil -f f -o srs [-w w] [-v] -c c
```

Get Device Status

```
tapeutil -f f -o gds [-w w] [-v]
```

Get Device Information

```
tapeutil -f f -o gdi [-w w] [-v]
```

Get Media Information

```
tapeutil -f f -o gmi [-w w] [-v]
```

Get Position

```
tapeutil -f f -o gpo [-w w] [-v] -t t
```

Set Position

```
tapeutil -f f -o spo [-w w] [-v] -t t -x x
```

Get Parameter

```
tapeutil -f f -o gpa [-w w] [-v] -t t
```

Set Parameter

```
tapeutil -f f -o spa [-w w] [-v] -t t -x x
```

Sync Buffer

```
tapeutil -f f -o syn [-w w] [-v]
```

Display Message

```
tapeutil -f f -o msg [-w w] [-v] -t t -y y1,y1
```

Note: Invoking *tapeutil* with the **-h** (*tapeutil -h*) or **-?** (*tapeutil -?*) flag displays the help information.

Flag Description

The supported flags, their meanings, their associated operations, and their acceptable ranges are as follows:

- ? Usage Help (stand-alone flag) {no value required}
- b Block Size (rea, wri)
{ $0 < (\text{block size} \times \text{blocking factor}) < 2097152$ }
- c Operation Count (eof, fsf, fsr, bsf, bsr, srs) {0–65535}
- d Destination Address (mov)
{device specific, determine range from Element Info}
- d1 Destination 1 Address (exc)
{device specific, determine range from Element Info}
- d2 Destination 2 Address (exc)
{device specific, determine range from Element Info}
- f Device Special File Name (always required)
{*/dev/rmt/1st* or similar}
- h Usage Help (stand alone flag) {no value required}
- m Multiples to Read or Write (rea, wri) {0–2097152}
- n Blocking Factor (rea, wri)
{ $0 > (\text{block size} \times \text{blocking factor}) < 2097152$ }
- o Operation (always required) {see the previous list}
- p The path to be enabled or disabled
- r Random Seed (wri) {0–65535}
- s Source Address (mov, pos)
{device specific, determine range from Element Info}
- t Type of Parameter Value
 - (gpo) {1=logical block, 2=physical block}
 - (spo) {1=logical block, 2=physical block}
 - (gpa) {1=block size, 2=compression, 3=buffering, 4=immediate, 5=trailer, 6=write protect, 7=acf mode, 8=capacity, 9=sili}
 - (spa) {1=block size, 2=compression, 3=buffering, 4=immediate, 5=trailer, 6=write protect, 8=capacity, 9=sili}
 - (msg) {1=display msg0, 2=display msg1, 3=flash msg0, 4=flash msg1, 5=alternate msg1/msg2}
 - (inq) {0=standard data, 1=page code}
 - {0=mt mode, 1=st mode}
- v Verbose Mode (optional for all commands, stand-alone flag)
{no value required, absence of flag means quiet mode}
- w Open Mode (optional for all commands)

- {1=read/write, 2=read only (default), 3=write only, 4=append}
- x** Parameter Value
- (lck) {1=lock, 2=unlock}
 - (lod) {1=load, 2=unload}
 - (spo) {0-65535}
 - (spa) {0-65535}
 - (inq) {0x0-0xFF}
 - (log) {0x00-0xFF}
 - (mod) {0x00-0xFF}
 - (rem) {1=prevent, 2=allow}
 - (ier) {0x0001-0xFFFF}
- y** Messages (msg) {message1,message2}
- z** Input/Output File Name
- (sdp) {path and name of the file in which to store dump}
 - (dmc) {path and name of the microcode image file}
 - (wri) {path and name of the file containing write data pattern}

Notes:

1. For read and write operations, the size of one buffer of data transferred during a single SCSI read or write command is determined by the product of the *Block Size* value and the *Blocking Factor* value. The number of these buffers transferred is determined by *Multiplier* value. The actual total number of bytes transferred is then (Block Size) x (Blocking Factor) x (Multiplier). If the device is set to fixed block mode (block size not equal to zero), the product of *Block Size* and *Blocking Factor* must be a multiple of the device block size setting.
2. For further information on the Get Parameter (*gpa*) and Set Parameter (*spa*) operations, see the STIOC_GET_PARM and STIOC_SET_PARM *ioctl* commands in the *IBM Ultrium Device Drivers: Programming Reference*, GC35-0483.

The following examples should help to demonstrate and clarify the command-line usage of the *tapeutil* program. For all examples, substitute the actual value of the special file associated with the target device.

- To query the serial number of the device:

```
tapeutil -f /dev/rmt/0m -o qsn -v
```
- To request inquiry data from the device:

```
tapeutil -f /dev/rmt/0m -o inq -v
```
- To request inquiry page data from the device:

```
tapeutil -f /dev/rmt/0m -o inq -t 1 -x 0x03 -v
```
- To request log sense page from the device:

```
tapeutil -f /dev/rmt/0m -o log -x 0x00 -v
```
- To request mode page from the device:

```
tapeutil -f /dev/rmt/0m -o mod -x 0x02 -v
```
- To move a cartridge from cell 32 to the tape drive (16) for a medium changer:

```
tapeutil -f /dev/rmt/4chng -o mo -s 32 -d 16 -v
```
- To write one hundred 64K blocks of data to the tape device:

```
tapeutil -f /dev/rmt/0mn -w 1 -o wri -b 65535 -n 1 -m 100 -v
```
- To write two file marks to the tape device:

```
tapeutil -f /dev/rmt/0mn -w 1 -o eof -c 2 -v
```

HP-UX Device Driver (ATDD)

- To rewind the tape device:
`tapeutil -f /dev/rmt/0mn -o rew -v`
- To read one hundred 64K blocks of data from the tape device:
`tapeutil -f /dev/rmt/0mn -o rea -b 65535 -n 1 -m 100 -v`
- To report the tape density:
`tapeutil -f /dev/rmt/0mn -o den -v`
- To prevent tape removal from the tape drive:
`tapeutil -f /dev/rmt/0mn -o rem -x 1 -v`
- To allow tape removal from the tape drive:
`tapeutil -f /dev/rmt/0mn -o rem -x 2 -v`
- To read the element status with the drive ID:
`tapeutil -f /dev/rmt/4chng -o dvc -v`
- To output the inventory information into a file:
`tapeutil -f /dev/rmt/4chng -o inv -t 1 > inv.txt`

Part 5. Linux Tape and Medium Changer Device Driver

Chapter 25. Introduction and Product Requirements

This chapter describes the IBM Linux Tape and Medium Changer Device Driver (IBMtape) for:

- IBM TotalStorage Ultrium External Tape Drive 3580
- IBM TotalStorage Ultrium Tape Autoloader 3581
- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582
- IBM TotalStorage Ultrium Scalable Tape Library 3583
- IBM TotalStorage UltraScalable Tape Library 3584

Purpose

The IBMtape and medium changer device driver is designed specifically to take advantage of the features provided by the IBM tape drives and medium changer devices. The goal is to give applications access to the functions required for basic tape operations (such as backup and restore) and medium changer operations (such as mount and demount the cartridges), as well as to the advanced functions needed by full tape management systems. Whenever possible, the driver is designed to take advantage of the device features transparent to the application.

Data Flow

The software described in this chapter covers the Linux Device Driver (IBMtape device driver) and the interface between the application and the tape device.

Figure 8 illustrates a typical data flow process.

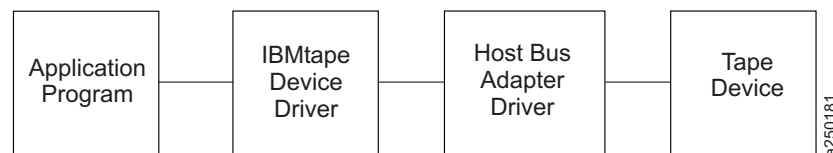


Figure 8. Data Flow Process

Product Requirements

More current information on supported hardware and software configurations for IBMtape will be in the README files on the distribution CD. The most current information is found in the README files on the IBM Device Drivers FTP site, which are located at <ftp://ftp.software.ibm.com/storage/devdrv/Linux>. Information specific to certain distributions are in the README files in the distribution-specific directories under the Linux directory.

Information there will include:

- Specifics on Linux distributions and kernel levels
- Supported Host Bus Adapter cards and required firmware and HBA device driver levels
- Other important information that is not included in this manual

Hardware Requirements for Intel Processors

One or more of the following processors is required by the IBMtape device driver:

- 32-bit Intel-compatible processors (uniprocessor or SMP) capable of running the Linux operating system.
- 64-bit Intel™ Itanium processors (uniprocessor or SMP) capable of running the Linux operating system.

One or more of the following IBM tape devices:

- IBM TotalStorage Ultrium External Tape Drive 3580
- IBM TotalStorage Ultrium Tape Autoloader 3581
- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582
- IBM TotalStorage Ultrium Scalable Tape Library 3583
- IBM TotalStorage UltraScalable Tape Library 3584

One or more of the following SCSI Host Bus Adapters:

- Adaptec 2940U2W (LVD)
- Adaptec 2944UW (HVD)
- Adaptec SCSI Adapter 29160 or IBM P/N 19K4646 for IBM xSeries® - Single Port LVD
- Adaptec SCSI Adapter 39160 - Dual Port LVD
- Adaptec SCSI Adapter 29320 - Single Port LVD
- Adaptec SCSI Adapter 39320 - Dual Port LVD

One or more of the following Fibre Channel Host Bus Adapters:

- QLogic QLA2200F, QLA2300F, QLA2310FL, QLA2340L, and QLA2342L Fibre Channel Adapters
- Emulex LP9002L, LP9802, and LP10000 Fibre Channel Adapters

Hardware Requirements for IBM @server pSeries Models

One or more of the following processors is required by the IBMtape device driver:

- IBM low, middle, and high range pSeries servers running a 64-bit Linux operating system.
- IBM POWER5 servers running a 64-bit Linux operating system.

One or more of the following IBM tape devices:

- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582 with Ultrium Generation II or higher generation drives
- IBM TotalStorage Ultrium Tape Scalable Library 3583 with Ultrium Generation II or higher generation drives
- IBM 3584 UltraScalable Tape Library Model L32 (base frame), D32 (expansion frame) with Ultrium Generation II or higher generation drives

One or more of the following host bus adapters:

- pSeries Feature Code 6228 Fibre Channel Adapter
- pSeries Feature Code 6239 Fibre Channel PCI-X Adapter
- pSeries Feature Code 5716 Fibre Channel PCI-X Adapter

- pSeries Feature Code 5712 PCI-X Dual Channel Ultra320 SCSI Adapter

Hardware Requirements for IBM [®]server zSeries Models

One or more of the following processors is required by the IBMtape device driver:

- IBM zSeries[®] 800 (z800) or zSeries 900 (z900) models running Linux for S390 (31-bit) or Linux for zSeries (64-bit)
- IBM zSeries 890 (z890) or zSeries 990 (z990) models running Linux for S390 (31-bit) or Linux for zSeries (64-bit)

One or more of the following IBM fibre channel tape devices:

- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582 with Ultrium Generation II or higher generation drives
- IBM TotalStorage Ultrium Tape Scalable Library 3583 with Ultrium Generation II or higher generation drives
- IBM 3584 UltraScalable Tape Library Model L32 (base frame), D32 (expansion frame) with Ultrium Generation II or higher generation drives

One or more of the following host bus adapters:

- FICON[™] card (feature 2315 or 2318) with Fibre Channel Protocol support
- FICON Express card (feature 2319 or 2320) with Fibre Channel Protocol support

Attention: Using a single Fibre Channel host bus adapter (HBA) for concurrent tape and disk operations is not recommended. Tape and disk devices require incompatible HBA settings for reliable operation and optimal performance characteristics. Under stress conditions (high I/O rates for tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems have been observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and using SAN zoning to minimize contention. IBM is focused on assuring server and storage configuration interoperability. IBM strongly recommends that your implementation plan includes provisions for separating disk and tape workloads.

Library Control Path and Tape Drive Data Path Failover Support

To use the library control path failover support, the feature code 1680 is required for the IBM 3582/3583/3584 library. Only these libraries with the native fibre IBM 3580 tape drives are supported by the Linux IBMtape control path failover device driver. For the availability of control path failover IBMtape device driver and supported Linux attachment environment, see the Ultrium ReadMe files on this FTP site: <ftp://ftp.software.ibm.com/storage/devdrv/Linux/>.

To use data path failover support on the tape drives, the feature code 1681 is required for the IBM Ultrium tape drives. Only the native fibre IBM 3580 Generation II or higher generation tape drives inside the IBM TotalStorage Ultrium libraries and IBM 3592 tape drives are supported by the Linux IBMtape data path failover device driver. For the availability of data path failover IBMtape device driver and supported Linux attachment environment, see the Ultrium ReadMe files on this FTP site: <ftp://ftp.software.ibm.com/storage/devdrv/Linux/>.

Software Requirements for Intel Processors

IBMtape device driver supports the following Linux distributions:

- Red Hat Enterprise Linux (32-bit and Itanium 2 64-bit kernels)
- SUSE LINUX Enterprise Server (32-bit and Itanium 2 64-bit kernels)

Software Requirements for IBM pSeries Models

The IBMtape device driver supports the following Linux distributions for Linux for pSeries (64-bit kernel):

- Red Hat Enterprise Linux
- SUSE LINUX Enterprise Server

Software Requirements for IBM zSeries Models

The IBMtape device driver supports the following Linux distribution for Linux for S390 (31-bit kernel) and Linux for zSeries (64-bit kernel):

- SUSE LINUX Enterprise Server

Chapter 26. Tape Drive, Media, and Device Driver Parameters

This chapter describes the parameters that control the operating modes of the tape drive, media, and device driver.

Configuration Parameters

The configuration parameters are used to set the operating mode of the tape drive and device driver when a device is opened. The installation defaults are provided for all parameters initially. Some of these parameters can be queried and set by *IBMtapeutil Query/Set Tape Parameters* on the menu. These parameters are kept on reopen, but are always restored back to the default values when the IBMtape device driver is reinstalled.

The nonchangeable configuration parameters are:

- Autoloading
- Capacity scaling
- Density Code
- Emulate autoloader
- Hook Word
- Logical write protect
- Maximum block size
- Minimum block size
- Medium type
- Read SILI bit
- Record space mode
- Volume ID for logging
- Write protect

The changeable configuration parameters are:

- Block size
- Buffered mode
- Compression
- Disable auto drive dump
- Disable SIM logging
- Logging
- Maximum SCSI transfer length
- Read past filemark
- Rewind immediate
- Trace
- Trailer labels

Nonchangeable Parameters:

The configuration parameters are used to set the operating mode of the tape drive and device driver when a device is opened. The nonchangeable parameters are detailed below.

Autoloading

This parameter enables the autoloading feature of the device driver. It is disabled by default and cannot be changed.

Capacity Scaling

This parameter sets the capacity or logical length of the current tape. By reducing the capacity of the tape, the tape drive can access data faster at the expense of data capacity. Capacity Scaling is not supported currently but might be supported in future releases of IBMtape.

Density Code

This parameter is the density setting for the currently loaded tape. Some tape devices support multiple densities and reports the current setting in this field. It cannot be changed by the application.

Emulate Autoloader

This parameter does not apply to IBM Ultrium devices and should be ignored.

Hook Word

This parameter is not supported in the IBMtape device driver.

Logical Write Protect

This parameter sets or resets the logical write protect of the current tape. This feature is not supported currently but might be supported in future releases of the IBMtape.

Maximum Block Size

This parameter is the maximum block size for the device.

Minimum Block Size

This parameter is the minimum block size for the device.

Medium Type

This parameter is the media type of the currently loaded tape. Some tape devices support multiple media types and reports different values in this field.

Read SILI Bit

SILI bit currently is not supported due to limitations associated with the Linux environment. SILI bit support might be enabled in future releases of the IBMtape.

Record Space Mode

This parameter specifies how the device driver operates when a forward or backward space record operation encounters a filemark. Only the SCSI mode is supported by IBMtape. When a forward or backward space record operation is

issued to the driver and a filemark is encountered, the device driver returns -1 and the *errno* variable is set to input/output error (EIO). On the forward space operation, the tape is left-positioned after the filemark (the end of tape side of the filemark). On the backward space operation, the tape is positioned before the filemark (the beginning of tape side of the filemark).

Volume ID for Logging

This parameter is the volume ID of the currently loaded tape. IBMtape device driver ignores this field.

Write Protect

This parameter, if set to TRUE, write protects the current mounted tape.

Changeable Parameters:

The configuration parameters are used to set the operating mode of the tape drive and device driver when a device is opened. The changeable parameters are detailed below.

Block Size

This parameter specifies the block size used for *read* and *write* operations. A value of zero means variable block size. Any other value is a fixed block size. The installation default is zero (variable length block size). See "Maximum SCSI Transfer Length" below for additional guidance.

Buffered Mode

This parameter specifies if the *read* and *write* operations should be buffered by the tape device. The default (recommended) value is TRUE.

Compression

Hardware compression is implemented in the device hardware. This parameter turns the hardware compression feature On and Off. If compression is enabled, effective performance can increase based on the compressibility of the data.

The installation default is On (use compression).

Disable Auto Drive Dump

This parameter is provided in the IBMtape version 1.2.2 or later. It is set to FALSE by default. If it is FALSE and the IBMtaped daemon is running and if an error occurs in the drive which creates a drive dump, the IBMtape device driver will automatically retrieve the drive dump and save it under the */var/log* directory by default. You can specify another directory in the */etc/IBMtaped.conf* file. See Chapter 28, "Configure and Run IBMtaped Daemon," on page 145 for details.

Disable SIM Logging

This capacity is not applicable to IBM Ultrium tape devices.

Logging (Volume Logging)

This parameter turns the volume information logging On and Off. With the IBMtape version 1.2.2 and later, the IBMtape device driver provides this support. It is set to On by default. If logging is On and the IBMtaped daemon is running, the IBMtape device driver will retrieve the full log sense data from the drive whenever

a tape is unloaded, or the drive reaches a log threshold. The log file will be saved in a binary format under the directory `/var/log` by default. You can specify another directory in `/etc/IBMtaped.conf` file. See Chapter 28, “Configure and Run IBMtaped Daemon,” on page 145 for details.

Note: This is volume logging, which is different from the error logging. IBMtape provides error logging whenever the IBMtaped daemon is running. See Chapter 28, “Configure and Run IBMtaped Daemon,” on page 145 for the details on error logging.

Maximum SCSI Transfer Length

By default, the maximum transfer length per device per SCSI command is 262,144 bytes (256 KB). Variable block read/write requests with a transfer length greater than the maximum transfer length will fail [*errno*: EINVAL]. When a fixed block size has been defined, large write requests are subject to both the granularity of the block size and the maximum transfer length. For example, with a fixed block size of 80,000 bytes and maximum transfer length of 262,144, a write request for 400,000 bytes (5 blocks of 80,000 each) is written to tape in two transfers. The first transfer is 240,000 bytes (3 blocks) and the second transfer is 160,000 (the remaining two blocks). You can increase the maximum transfer length to enhance the data throughput. This can be done either using *IBMtapeutil*, option 48 – **Query/Set Tape Parameters**, or a customized STIOCSETP input/output control (*ioctl*) call. However, setting the transfer length greater than the default 256 KB does not guarantee a noticeable increase in data throughput. A maximum transfer length of 256 KB is highly recommended.

Read Past Filemark

If this parameter is set to TRUE, when a *read* operation encounters a filemark, it returns the number of bytes read before encountering the filemark and positions the tape head after the filemark. If the *read_past_filemark* parameter is set to FALSE, when a *read* operation encounters a filemark, and if data was read, the *read* function returns the number of bytes read, and positions the tape head before the filemark. If no data was read, the *read* returns 0 bytes read and positions the tape head after the filemark.

This installation default is FALSE.

Rewind Immediate

This parameter turns the immediate bit On and Off for rewind commands. If it is set to On, the rewind tape operation executes faster, but the next command takes a long time to finish unless the physical rewind operation has completed. Setting this parameter reduces the amount of time it takes to close a device for a Rewind on Close special file.

The installation default is Off (no rewind immediate).

Trace

This parameter turns the trace facility On and Off. With the IBMtape version 1.2.2 and later, the IBMtape device driver provides this support. It is set to On by default. If trace is On and the IBMtaped daemon is running, the IBMtape device driver will retrieve the trace from the driver if trace level is set to 1 or 2 in the `/etc/IBMtaped.conf` file. The trace file will be saved under the directory `/var/log` by default. You can specify another directory in `/etc/IBMtaped.conf` file. See Chapter 28, “Configure and Run IBMtaped Daemon,” on page 145 for details.

Trailer Labels

If this parameter is set to On, writing records past the early warning mark on the tape is allowed. The first *write* operation, after detecting the early warning mark, fails and the *errno* variable is set to ENOSPC. No data is written during the operation. All subsequent *write* operations are allowed to continue until the physical end of the volume is reached and input/output error *errno* (EIO) is returned.

If this parameter is set to Off, writing records past the early warning mark is not allowed. *Errno* variable is set to ENOSPC.

The installation default is On (with trailer labels).

Linux Device Driver (IBMtape)

Chapter 27. Installation and Configuration Instructions

The IBMtape device driver for Linux is an installable kernel module, supplied in an *rpm* package. The utility tools for IBMtape are supplied in a *tar* file. The following sections describe installation, removal, configuration, and verification procedures for IBMtape and its utility tools. Refer to Linux documentation for *rpm* and *tar* command information. You must have *root* authority to proceed with the installation of the driver.

Conventions Used

In the subsequent pages, you will see file names with *xxx* or *x.x.x* in them. These refer to the version of the driver, which change as IBM releases new driver levels. Use the actual driver version numbers as you perform the procedures. If there is more than one version on the installation CD, use the latest version.

Commands that you are to type are indicated with a leading ">" character, which indicates the shell prompt. Some of the commands have a "—some text" string after them, for example, ">cd IBMtapeutil.x.x.x.i386 — to change the directory". The "—some text" items are explanatory comments to the reader and should not be specified when you enter these commands, for example, "cd IBMtapeutil.x.x.x.i386" is all you would input on the previous example.

When you are installing from the *tar* packages, you need to copy the *tar* files from the installation CD to a directory on your hard drive. It is recommended that you keep the *tar* files in that directory, but it isn't necessary. When you unpack the *tar* files, you need to keep the directories that were created and the files that the installation process created in those directories. You also need to remember the name of the top level directory in order to uninstall and/or update the utilities.

Components Created During Installation

The IBMtape package consists of the device driver and a number of associated files. Components created during IBMtape installation are listed in the following table:

Linux Device Driver (IBMtape)

Table 12. Components Created During IBMtape Installation

Component	Description
<code>/lib/modules/(Your system's kernel name)/kernel/drivers/scsi/IBMtape.o</code>	Device driver module for current kernel version
<code>/usr/bin/IBMtaped</code>	IBMtaped daemon
<code>/usr/include/sys/IBM_tape.h</code>	Device driver header file
<code>/etc/IBMtaped.conf</code>	IBMtaped daemon configuration file
<code>/usr/share/doc/IBMtape-xxx/IBMtape_Ultrium.Readme</code> (for Red Hat) <code>/usr/share/doc/packages/IBMtape/IBMtape_Ultrium.Readme</code> (for SUSE LINUX)	Readme file for IBMtape using IBM Ultrium devices
<code>/usr/share/doc/IBMtape-xxx/IBMtaped.Readme</code> (for Red Hat) <code>/usr/share/doc/packages/IBMtape/IBMtaped.Readme</code> (for SUSE LINUX)	Readme file for IBMtaped daemon
<code>/usr/share/doc/IBMtape-xxx/License</code> (for Red Hat) <code>/usr/share/doc/packages/IBMtape/License</code> (for SUSE LINUX)	License documentation for IBMtape

The IBMtapeutil package consists of *IBMtapeutil*, *IBMtapeconfig*, and the source files for *IBMtapeutil*.

Table 13. Components Created During IBMtapeutil Installation

Component	Description
<code>/usr/bin/IBMtapeconfig</code>	Utility to create special files
<code>/usr/bin/IBMtapeutil</code>	Utility and service aid program
<code>IBMtapeutil.h</code>	Header file for IBMtapeutil
<code>IBMtapeutil.c</code>	Source program for IBMtapeutil
<code>IBMtapeutil.ReadMe</code>	ReadMe file for IBMtapeutil
<code>makefile</code>	Makefile for IBMtapeutil

Installation Procedure

If IBMtape is already installed on your system, see the next section for the update procedure. This section assumes that you are installing the IBMtape device driver onto a system where it is not installed.

If you are installing IBMtape on a system running Linux for S390 or Linux for zSeries, you need make sure that the OpenFCP adapter device driver `zfcplib` is loaded in the kernel. See “Configuring Tape and Medium Changer Devices on IBM @server zSeries Models” on page 141 in this chapter for how to configure and install `zfcplib`.

If you have the installation CD, use the following instructions to install the device driver:

1. Place the CD into the CD-ROM drive on your Linux system.
2. If you are using X Window System, double-click the CD-ROM icon on the screen and open the Linux folder. Otherwise, mount the CD-ROM and change to directory `/mnt/cdrom/Drivers/Linux` on the command line.
3. Consult the *Readme* file on the CD for any important information pertaining to the device driver. Information in this file takes precedence over information in the manual.

4. Change into the directory under `/mnt/cdrom/Drivers/Linux` which supports your particular Linux distribution. Consult the Readme file in the distribution directory.
5. Assuming that your system is running a Linux kernel on a 32-bit Intel-compatible processor, then use the following command:


```
> rpm -i IBMtape.x.x.x.i386.rpm --- to install
```
6. Copy the utilities `tar` file from the installation CD to a directory on your hard drive that is kept. If you used the `tar` command to unpack the driver, you can use the same directory for that `tar` file. Change to that directory and issue the following commands:


```
> tar -xvf IBMtapeutil.x.x.x.tar --- unpack the files from the archive
                                     (x.x.x is the version)
> cd IBMtapeutil.x.x.x --- change to the utilities directory just
                                     created by the tar
> make install --- install the utilities
```
7. Unmount the CD-ROM. Remove the CD from the CD-ROM drive and store it in a safe place.

Updating Procedure

Before using this procedure to update your device driver to a different level, use the following command to obtain your existing IBMtape device driver version, if there is at least one tape device attached to your system:

```
> IBMtapeutil -f /dev/IBMtape0 qryversion
```

You can uninstall the driver first, then install the newer version. For example:

```
>rpm -e IBMtape
>rpm -i IBMtape.x.x.x.i386.rpm
```

Note that all tape devices that use the IBMtape device driver must be closed and can not be in use when IBMtape is uninstalled.

Querying Installed Package

The installed `rpm` package can be queried by running the following commands to display information associated with the package.

To display information about IBMtape:

```
>rpm -qi IBMtape
```

To display the package file list, enter the command:

```
>rpm -ql IBMtape
```

To display the states of files in the package, i.e. *normal*, *not installed* or *replaced*:

```
>rpm -qs IBMtape
```

Verifying Installation/Updating

You can run the following command to verify your IBMtape device driver version if there is at least one tape device attached to your system:

```
IBMtapeutil -f /dev/IBMtape0 qryversion
```

If the IBMtape device driver is installed from the `rpm` package:

```
rpm -V IBMtape
```

Configuring Tape and Medium Changer Devices on Intel-Compatible Systems

If you have not physically attached your tape and medium changer devices to your Linux server, perform the following steps:

1. Halt the Linux server.
2. Power down the Linux server.
3. Physically attach the tape and medium changers to the host bus adapter cards. Consult your hardware documentation for details.
4. Power on the tape and medium changer devices. Verify the initialization process has completed.
5. Power on and boot the Linux server.

Note: Hot plugging SCSI devices while the Linux server is running can cause hardware damage.

After the driver software is installed and a tape device is connected to the adapter, the device can be configured and made available for use. Access to the device is not provided until the device is configured.

If your system is attached to an IBM 3583 Ultrium Scalable Tape Library with the integrated router, before installing the Qlogic driver set the host type of the router to `solaris` and make sure that the logical unit numbers of the control unit, medium changer, and the connected tape drives are contiguous (otherwise, the Qlogic device driver will not recognize all of the attached devices). To view the LUNs of attached devices, log onto the router and use the `fcShowDevs` command. If the LUNs are not contiguous, use the `mapCompressDatabase` command to delete the invalid LUNs and make the valid LUNs contiguous. If you have the StorWatch™ Specialist installed, you can use the Specialist to do this configuration. For further information about the StorWatch Specialist, see the *IBM Storage Area Network Data Gateway Installation and User's Guide*. You can download the guide from the Web at:

<http://www.storage.ibm.com/hardsoft/products/sangateway/support/cdr/Document/sdgd.doc.htm>

If you are running the IBMtape device driver, version IBMtape.1.4.1 or higher, and the IBMtaped daemon is running, IBMtaped will automatically create the special files under the `/dev` directory for you. Otherwise, you need to run *IBMtapeconfig* to manage the creation of special files for the attached devices:

```
> IBMtapeconfig
```

IBMtapeconfig removes all the existing `/dev/IBMtape` and `/dev/IBMchanger` special files, and then creates new ones based on the information in `/proc/scsi/IBMtape` and `/proc/scsi/IBMchanger`.

Note: If a new device is attached while the system is on, reboot the system first, then run *IBMtapeconfig*.

To configure the tape devices, use the *IBMtapeutil* application program. Choose **Query/Set Tape Parameters** in the menu.

Configuring Tape and Medium Changer Devices on IBM [®]server pSeries Models

Follow the same instructions as documented in the previous section. You will need to configure the Emulex Linux device driver if you have fibre channel tape devices attached to your pSeries system. For instructions on how to configure the Emulex device driver, see the `IBMTape_Ultrium_pSeries.ReadMe` at <ftp://ftp.software.ibm.com/storage/devdrv/Linux/>.

Configuring Tape and Medium Changer Devices on IBM [®]server zSeries Models

The fibre channel topology supported for the zSeries models is fabric only. Neither point-to-point connection nor arbitrated loops are supported by the current zSeries Fibre Channel Protocol. See the document "Getting Started with zSeries Fibre Channel Protocol" for more details on the supported configurations for fibre channel device attachment on zSeries models. This document can be downloaded from this Web site:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp0205.pdf>

The Linux fibre channel adapter device driver `zfcplib` is available in the kernel that supports zSeries Fibre Channel Protocol. The `zfcplib` device configuration methods in 2.6 and 2.4 kernels are different. For 2.6 kernels, see Chapter 5 in the document entitled "Linux for s390 and zSeries Device Drivers, Features, and Commands," published November 30, 2004. You can download this document from the following Web site:

<http://awlinux1.alphaworks.ibm.com/developerworks/linux390/docu/1x26apr04dd03.pdf>

For 2.4 kernels, there are three ways to load the `zfcplib` device driver in order to see the attached tape devices.

1. Create a `/etc/zfcplib.conf` file and make a ramdisk to statically attach tape devices into your system. You can use this method only if you have a persistent mapping in a SAN environment. Every time you reboot the system, the `zfcplib` will be automatically loaded and the tape devices can be seen from the system.
2. Modify the `/etc/modules.conf` file to add the `zfcplib` module parameters; then run the `"depmod -A"` and `"modprobe zfcplib"` command. Do not use this choice together with the first one, otherwise it will cause conflicts. The `zfcplib` map in `/etc/modules.conf` always takes higher priority than the map in `/etc/zfcplib.conf`.
3. Run the `"modprobe zfcplib"` command first, then dynamically add a tape device into the system after you physically attach a fibre channel tape device to the switch.

Use `/etc/zfcplib.conf` File

First, you need to add the device map into this file. The following is an example of `zfcplib.conf`:

```
0xf1c0 0x1:0x5005076300402733 0x0:0x0000000000000000;\
0xf1c1 0x1:0x5005076300402733 0x0:0x0001000000000000
```

The `zfcplib` device driver uses the `"map"` module parameter to recognize a physically attached tape device. `"map"` takes the following format:

```
map="<devno><port scsi-id>:<wwpn><unit-scsi-lun>:<fcp-lun>;...."
```

where:

Linux Device Driver (IBMtape)

devno device number of the host bus adapter (16 bits, see `/proc/subchannels`). It is "0xf1c0" or "0xf1c1" in the above example.

port scsi-id

Linux internal SCSI ID assigned to the Fibre Channel port of the SCSI target device (32-bit, must not be 0, must be a unique one-to-one mapping for each World Wide Port Name. It is "0x1" in the above example.

wwpn World Wide Port Name identifying the Fibre Channel port of the SCSI target device (64-bit). It is "0x5005076300402733" in the above example.

unit scsi-lun

Linux internal SCSI Logical Unit Number (32-bit). It is "0x0" in the above example.

fcp-lun

Logical Unit Number associated with the SCSI target device (64-bit). In the above example, "0x0000000000000000" is the Logical Unit Number 0, and "0x0001000000000000" is the Logical Unit Number 1.

We recommend, for tape attachment, that each logical unit number be associated with a unique devno. If you use the same devno numbers for several logical units, you should ensure that each `<unit-scsi-lun>` is unique.

After `/etc/zfc.conf` is created, run the following commands:

```
>mk_initrd  
>zipl
```

Then reboot the system. After it is booted up, your tape device should be shown in `/proc/scsi/scsi` file.

Modify the `/etc/modules.conf` File

You can add tape device mapping into `/etc/modules.conf` if you don't want to use `/etc/zfc.conf`. The following example demonstrates the zfc configuration in `/etc/modules.conf`:

```
options zfc map="\n0xf1c0 0x1:0x5005076300402733 0x0:0x0000000000000000;\n0xf1c1 0x1:0x5005076300402733 0x0:0x0001000000000000"
```

The map arguments are the same as the ones listed in for the `/etc/zfc.conf` file. where:

After modifying the `/etc/modules.conf` file, save and close it. Then run the following command:

```
>depmod -A  
>modprobe zfc
```

This will install the zfc device driver and all of its prerequisite kernel modules. Now you can check the file `/proc/scsi/scsi` to see if all of the attached tape devices are shown in this file. If not, then check the fibre channel connection, such as the fibre cables, or if the devices are powered on, etc. Then run the following commands to install zfc:

```
>rmmod zfc  
>modprobe zfc
```

Dynamically Attaching a Tape Device

If you physically attach a tape device on the switch and `zfc` is already loaded, you do not need to reboot the Linux system in order to add this entry in the `/proc/scsi/scsi` file. The `zfc` device driver provides an "add_map" proc system entry under the directory `/proc/scsi/zfc` to allow you to dynamically add the device into the system. For example, to add two logical units from the above example into the system, you can issue the following commands:

```
> echo "0xf1c0 0x1:0x5005076300402733 0x0:0x0000000000000000;\
0xf1c1 0x1:0x5005076300402733 0x0:0x0001000000000000" > /proc/scsi/zfc/add_map
> echo "scsi add-single-device 0 0 1 0" > /proc/scsi/scsi
> echo "scsi add-single-device 1 0 1 1" > /proc/scsi/scsi
```

The "scsi add-single-device" takes four parameters, corresponding to the four parameters "scsi", "Channel", "Id", and "Lun" in the `/proc/scsi/scsi` file. The value of "scsi" is 0 for the first devno, 1 for the second devno (if it is different from the first devno), etc. The value of "Channel" can start from 0 for each different "scsi" value. The value of "Id" is the one you use for `<unit scsi-lun>` in the above mapping. The value of "Lun" is the logical unit number of the target device, for example, the last number in the above mapping.

Currently, the `zfc` device driver does not support dynamically removing the attached devices. If you need to remove the tape devices from the system, do "rmmod `zfc`". Then you can delete the entry in `/etc/modules.conf` and reload `zfc`, or reload `zfc` first and dynamically add the devices you want.

After you have done all the mapping, if you can see all of the attached tape devices in `/proc/scsi/scsi`, you have successfully attached those devices to your system. Next you can install the IBMtape device driver. See the "Installation Procedure" section in this chapter for the instructions on how to install IBMtape.

After the IBMtape device driver is installed, you can run `IBMtapeconfig` to create special files for the tape devices. `IBMtapeconfig` is provided in the `IBMtapeutil` tar package. After running `IBMtapeconfig`, you will be able to see the special files under the `/dev` directory for the attached tape devices.

Uninstall Procedure

Note: All tape devices that use the IBMtape driver must be closed and cannot be in use when IBMtape is uninstalled or the uninstall fails.

Run the following command:

```
> rpm -e IBMtape          --- to remove
```

Linux Device Driver (IBMtape)

Chapter 28. Configure and Run IBMtaped Daemon

Starting with IBMtape version 1.2.5, the IBMtape device driver provides an error diagnostic daemon (IBMtaped) which provides the following capabilities:

1. Full error logging and tracing of the IBMtape device driver
2. When drive dumps, log sense data, and/or SIM/MIMM error information are created by the tape drive, the daemon will automatically retrieve that data and save it to the hard drive on your Linux system.

Because IBMtaped requires a minimal amount of system resource and because it provides these necessary diagnostic capabilities, IBM recommends that you leave the daemon enabled at all times.

Install IBMtaped

IBMtaped is automatically installed at `/usr/bin/IBMtaped` when you install the IBMtape device driver using the `rpm` package. See Chapter 27, “Installation and Configuration Instructions,” on page 137 for instructions on installing the IBMtape device driver.

Configure IBMtaped

You can customize the operation of IBMtaped by modifying its configuration file, which is located at `/etc/IBMtaped.conf`. The daemon only reads the configuration file when it starts; consequently, if you make modifications to the configuration file, stop the daemon and restart it so that your updates will be recognized by the daemon.

Tracing

Three levels of tracing are supported for the IBMtape device driver and are defined as follows:

- 0** With tracing set to 0, very minimal tracing is recorded from the IBMtape device driver.
- 1** With tracing set to 1, IBMtaped records information associated with each `ioctl` called. If a device error occurs and SCSI sense data is obtained from the device, a subset of that sense data will also be recorded. This is the default setting for tracing.
- 2** With tracing set to 2, IBMtaped will record tracing messages for each SCSI command. If a device error occurs and SCSI sense data is obtained from the device, all sense data will also be recorded. This tracing level should only be used when a specific problem is being diagnosed due to the potential for huge amounts of data being generated.

Set the `IBMtapeTrace` variable in the `/etc/IBMtaped.conf` file to 0, 1, or 2, depending on what level of tracing you desire. If the `IBMtapeTrace` variable is set to an invalid number, the IBMtaped daemon will not start.

Tracing information is written to a file named `/var/log/IBMtape.trace`, by default. Information is written into the file until it is 1 MB in size, by default. After 1 MB of information is written, the file is archived (using the Linux `ar` command) into file

Linux Device Driver (IBMtape)

IBMtape.a in the same directory. In the archive, the filename will be renamed to *IBMtape.trace.timestamp*, where *timestamp* reflects the time that the file was archived.

You can change the directory to which the tracing information is written or the default maximum size of the trace file by modifying settings in the *IBMtaped.conf* file. See the instructions in the *IBMtaped.conf* file for details.

Error Logging

IBMtaped records certain error messages from the IBMtape device driver in a file named */var/log/IBMtape.errorlog*, by default. Information is written into the file until it is 1 MB in size, by default. After 1 MB of trace information is written, the file is archived (using the Linux *ar* command) into file *IBMtape.a* in the same directory. In the archive, the filename will be renamed to *IBMtape.errorlog.timestamp*, where *timestamp* reflects the time that the file was archived.

You can change the directory to which the error logging information is written or the default maximum size of the error log file by modifying settings in the *IBMtaped.conf* file. See the instructions in the *IBMtaped.conf* file for details.

Whenever the IBMtaped daemon is running, error logging is enabled if tracing is enabled. Following is an example an error log record:

```
IBMtape0---E0001 Tue Sep 10 14:04:57 2002
  Scsi Path   : 03 00 00 00
  CDB Command : 01 00 00 00 00 00
  Status Code : 08 00 00 01
  Sense Data  : 70 00 04 00 00 00 00 58 00 00 00 00 00 00 FF 0B
                C4 77 00 00 00 06 01 40 00 00 00 00 00 00 01 00
                10 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  Description : Hardware Error
```

The first line indicates the tape device special file name and the device serial number, and the timestamp when the error message was recorded. "Scsi Path" is the SCSI path for this logical unit. It matches the order of the *scsi/Channel/Id/Lun* information in the */proc/scsi/scsi* file. "CDB Command" is the command data block of the SCSI command. "Status Code" is the returned result from the Linux SCSI middle layer device driver (*scsi_mod.o*). The four bytes represent *driver_byte*, *host_byte*, *msg_byte*, and *status_byte*, respectively. "Sense Data" is the full SCSI sense data returned from the target. "Description" is a person-readable text string obtained by parsing the sense key field of the sense data.

The following circumstances are not logged in the *IBMtape.errorlog* file:

1. Sense key is 0, and the sense data indicates an overlength or an underlength read, or encountering a file mark or the end of data
2. Sense key is 2, and the ASC/ASCQ indicates the device is becoming ready
3. Sense key is 6, indicating a unit attention
4. Sense key is 8, and the ASC/ASCQ indicates the end of data

Volume Logging

The IBMtape device driver retrieves the full log sense data from the tape drive whenever the drive reaches a log threshold, or a tape is unloaded from the drive, or the drive is reset through an application. This data is stored in binary in a file named *IBMtape.timestamp.log*, where *IBMtape* is the device special file (for example, *IBMtape1*, *IBMtape2*, etc.) and *timestamp* reflects the time the file was created. Each time log sense data is obtained, it is written to a new file. Use the appropriate tape drive hardware reference manual to decode the log sense data.

The volume logging data is stored in the */var/log* directory by default. You can specify another directory in the */etc/IBMtaped.conf* file.

There are two configuration parameters in the */etc/IBMtaped.conf* file that you can tailor to affect the number of log sense files that are kept on your system: *IBMtapeMaxLogSenseFiles* and *IBMtapeAutoLogSenseFileOverWrite*. *IBMtapeMaxLogSenseFiles* can be 0 or a positive decimal number. *IBMtapeAutoLogSenseFileOverWrite* can be 0 or 1. If *IBMtapeMaxLogSenseFiles* is 0, *IBMtapeAutoLogSenseFileOverWrite* is ignored, and each time log sense data is obtained, it is written to a new file. If *IBMtapeMaxLogSenseFiles* is a positive number and *IBMtapeAutoLogSenseFileOverWrite* is 0, each time log sense data is created, IBMtaped will write that data to a file until *IBMtapeMaxLogSenseFiles* have been created; then IBMtaped will stop creating new files, even if new log sense data is produced. If *IBMtapeMaxLogSenseFiles* is a positive number and *IBMtapeAutoLogSenseFileOverWrite* is 1, each time log sense data is created, IBMtaped will write that data to a file until *IBMtapeMaxLogSenseFiles* have been created; then when new log sense data is detected, IBMtaped will delete the oldest log sense file and create a new file with the newest log sense data; thus, only the newest data will be kept.

By default, *IBMtapeMaxLogSenseFiles* is 0 and *IBMtapeAutoLogSenseFileOverWrite* is 1, which means that every time log sense data is created, it will be written to a new file.

Automatically Retrieve a Drive Dump

If a condition occurs in the drive such that a drive dump is created, IBMtaped will retrieve the drive dump and save it in a file named *IBMtape.timestamp.dmp*, where *IBMtape* is the device special file (for example, *IBMtape1*, *IBMtape2*, etc.) and *timestamp* reflects the time the file was created. Each time a drive dump is obtained, it is written to a new file. IBM service might request that you forward drive dumps to them for analysis.

The drive dumps are stored in the */var/log* directory by default. You can specify another directory in the */etc/IBMtaped.conf* file.

There are two configuration parameters in the */etc/IBMtaped.conf* file that you can tailor to affect the number of drive dumps that are kept on your system: *IBMtapeMaxDumpFiles* and *IBMtapeAutoDriveDumpFileOverWrite*. *IBMtapeMaxDumpFiles* can be 0 or a positive decimal number. *IBMtapeAutoDriveDumpFileOverWrite* can be 0 or 1. If *IBMtapeMaxDumpFiles* is 0, *IBMtapeAutoDriveDumpFileOverWrite* is ignored, and each time a drive dump is obtained, it is written to a new file. If *IBMtapeMaxDumpFiles* is a positive number and *IBMtapeAutoDriveDumpFileOverWrite* is 0, each time a dump is obtained, IBMtaped will write that data to a file until *IBMtapeMaxDumpFiles* have been created; then IBMtaped will stop creating new files, even if new drive dumps are produced. If *IBMtapeMaxDumpFiles* is a positive number and

Linux Device Driver (IBMtape)

IBMtapeAutoDriveDumpFileOverWrite is 1, each time a dump is obtained, IBMtaped will write that data to a file until *IBMtapeMaxDumpFiles* have been created; then when a new drive dump is detected, IBMtaped will delete the oldest drive dump file and create a new file with the newest drive dump data; thus, only the newest data will be kept.

By default, *IBMtapeMaxDumpFiles* is 0 and *IBMtapeAutoDriveDumpFileOverWrite* is 1, which means that every time a drive dump is obtained, it will be written to a new file.

Selective Tracing

IBMtape provides facilities by which you can disable and enable tracing, error logging, auto-retrieving drive dumps, and auto-retrieving SIM/MIM data. You can selectively enable or disable them through the *IBMtapeutil* Query/Set Tape Parameters operation or through an application program that uses the *STIOC_SETP ioctl*. These settings persist until the device driver is restarted, or the host system is rebooted. The parameters and their definitions are as follows:

trace This parameter is set to On by default, which enables IBMtape tracing of activities and error logging on a particular tape drive. Set this parameter to off to stop tracing and error logging.

logging

This parameter is set to On by default and enables logging of log sense data. Setting this flag to Off suppresses volume logging for this device.

disable_sim_logging

This parameter does not apply to the IBM Ultrium tape devices.

disable_auto_drive_dump

This parameter controls the saving of drive dumps for a device. By default it is set to Off which causes drive dumps to be saved. Set this flag to On to suppress the saving of drive dumps.

Run IBMtaped

If you are running the IBMtape device driver, version 1.4.1 or higher, after installing IBMtape IBMtaped will start running even if your system does not have a tape device attached. If you add a new tape device into your Linux system, IBMtaped will automatically create a special file under the */dev* directory.

If you are running the IBMtape device driver, version 1.3.x or less, IBMtaped will not automatically start if there is no tape device attached. After you attach a new tape device, you'll need to start the IBMtaped daemon.

You can invoke IBMtaped from the command line. IBMtaped takes zero or more of the parameters as listed below:

```
IBMtaped [start stop restart status]
```

IBMtaped or IBMtaped start

Starts the daemon. If there is already an IBMtaped running, the new one will be aborted. (Use "IBMtaped restart" if IBMtaped is already running.)

IBMtaped stop

Terminates the daemon and frees all the resources associated with the daemon. When the daemon is stopped, no information is saved.

IBMtaped restart

Terminate the currently running daemon and starts a new one. The new

daemon will read the */etc/IBMtaped.conf* file. This command should be used after modifying the */etc/IBMtaped.conf* file while IBMtaped is running.

IBMtaped status

Prints a message on stdout indicating whether the daemon is running or not.

Note: If you run the "rmmod IBMtape" command to remove the IBMtape device driver from the running kernel, you need to stop the IBMtaped daemon first; otherwise you will get a "Device or Resource Busy" error.

Linux Device Driver (IBMtape)

Chapter 29. Special Files

After the driver is installed and a device is configured and made available for use, access is provided through the special files. These special files, which consist of the standard Linux special files for devices, are in the */dev* directory.

Special Files for the Tape Device

Each tape device has a set of special files that provide access to the same physical drive, but provide different attributes. The table below shows the attributes of the special file.

Note: The asterisk (*) in *IBMtape** represents a number assigned to a particular device, such as *IBMtape0*.

For tape drives with attached medium changer devices, the *IBMchanger** special file provides a separate path for issuing commands to the medium changer. When this special file is opened, the application can view the medium changer as a separate device. Both the tape and changer special file can be opened at the same time.

Table 14. Special Files for IBM Ultrium Tape Devices

Special File Name	Rewind on Close
<i>/dev/IBMtape*</i>	Yes
<i>/dev/IBMtape*n</i>	No

Special Files for the Medium Changer Device

After the driver is installed and a medium changer device is configured and made available for use, access to the robotic device is provided through the *IBMchanger* special file in the */dev* directory. The asterisk (*) represents a number assigned to a particular device, such as *IBMchanger0*. The term *IBMchanger* is used for a SCSI medium changer device. The *IBMchanger** special file provides a path for issuing commands to control the medium changer robotic device.

The file descriptor that results from opening the *IBMchanger* special file does not support the following operations:

- Read
- Write
- Open in Append mode
- Commands designed for a tape device

Linux Device Driver (IBMtape)

Chapter 30. Alternate Pathing Support for Library Control Path Failover

Note: The library control path failover feature code must be installed prior to enabling the alternate pathing support in the Linux IBMtape device driver. Refer to Table 2 on page 12 to determine which feature code is required for your machine type.

The Linux IBMtape device driver alternate pathing support will configure multiple physical control paths to the same logical library within the device driver and provide automatic failover to an alternate control path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) which are connected to a library with two drive control ports enabled. The first HBA is connected to the first control port drive, and the second HBA is connected to the second control port drive. This connection provides two physical control paths to the same library for redundancy if one path from an HBA to the library fails.

When the host bus adapter device drivers are loaded into the running system, each HBA detects a control port to the same library, and two logical devices can be created (for example, `/dev/IBMchanger0` and `/dev/IBMchanger1`) by the IBMtaped daemon or the *IBMtapeconfig* script. Each logical device is a physical path to the same library. An application can open and use only one logical device at a time, either `IBMchanger0` or `IBMchanger1`, because they represent the same physical device. Without the IBMtape alternate pathing support, if an application opens `IBMchanger0` and a permanent path error occurs (because of an HBA, cable, or drive control port failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (`IBMchanger1`), but failing transaction has to be restarted from the beginning.

When the alternate pathing support is enabled, the device driver configures them internally as a single device with multiple paths. The application can still open and use only one logical device at a time (either `IBMchanger0` or `IBMchanger1`). If an application opens `IBMchanger0` and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (`IBMchanger1`). If successful, the current operation continues on the alternate path without interrupting the application.

Configuring and Unconfiguring Alternate Pathing Support

Alternate pathing support is not enabled automatically when the device driver is installed. The Linux IBMtape device driver provides a driver parameter *alternate_pathing* for you to enable the library control path failover.

To enable the failover support in the IBMtape device driver software, you need do the following steps after installing the IBMtape *rpm* package:

```
>IBMtaped stop (stop the IBMtaped daemon)
>rmmmod IBMtape (unload the IBMtape driver from the memory)
```

Add the following line in your `/etc/modules.conf` file for 2.4 kernels or `/etc/modprobe.conf.local` file for 2.6 kernels:

Linux Device Driver (IBMtape)

```
options IBMtape alternate_pathing=1
>depmod
>modprobe IBMtape (re-load the IBMtape driver into memory)
>IBMtaped (re-start IBMtaped daemon)
```

You can ignore the "Unresolved symbols in /lib/modules/<your kernel name>/drivers/scsi/IBMtape.o" message after the "depmod" command. You can check if the IBMtape driver has recognized multiple control paths for your library by reading the /proc/scsi/IBMchanger file.

```
>cat/proc/scsi/IBMchanger
```

If your library lists "Primary" or "Alternate" under "FO Path", you have successfully enabled control path failover feature for your library. If it is "NA" listed under "FO Path", then the control path failover is not enabled.

After the alternate pathing support is enabled, it remains set until the IBMtape driver is reloaded with the *alternate_pathing* driver parameter set to OFF . The alternate pathing setting is retained even if the system is rebooted. If you want to turn off the control path failover feature in the IBMtape device driver, you can perform the following steps:

```
>IBMtaped stop
>rmmmod IBMtape
```

Delete the following line in your /etc/modules.conf file:

```
options IBMtape alternate_pathing=1
>depmod
>modprobe IBMtape
>IBMtaped
```

Primary and Alternate Paths

When IBMtape is loaded into kernel memory, the first logical medium changer device that IBMtape sees in the system will be the primary path for that medium changer. The other logical medium changers that IBMtape attached for the same medium changer will be configured as alternate paths. The device driver supports up to 16 physical paths for a single device.

The primary and alternate path information can be obtained by the following command:

```
>cat /proc/scsi/IBMchanger
```

The following is an example of a /proc/scsi/IBMchanger file:

```
IBMtape version: 1.4.1
IBMtape major number: 253
Attached Changer Devices:
```

Number	Model	SN	HBA	FO Path
0	ULT3583-TL	IBM1234567	QLogic Fibre Channel 2200	Primary
1	ULT3583-TL	IBM1234567	QLogic Fibre Channel 2300	Alternate
2	ULT3583-TL	IBM1234567	QLogic Fibre Channel 2300	Alternate

The labeling of a logical device as either a primary or alternate path is for information only, in order to:

1. Be able to identify the actual number of physical devices configured on the system and a specific logical device associated with them. There will be only

one logical device labeled as the primary path for each physical device. However, there can be multiple logical devices labeled as an alternate path for the same devices.

2. Provide information about which logical devices configured on the system have alternate pathing support enabled.

Querying Primary and Alternate Path Configuration

You can display the primary and alternate path configuration for all devices by reading the `/proc/scsi/IBMchanger` file, as explained in the above section.

You can also display the primary and alternate path configuration for any device by running the following command:

```
>IBMtapeutil -f /dev/IBMchangerx path
      (IBMchangerx is the logical name of any device)
```

This command shows specific information for the primary path and all alternate paths, such as the logical name of the device, the attached host bus adapter, the channel id, the target id, the logical unit number under the target, the current enabled status, and how many paths are configured for the device.

Disable and Enable Primary and Alternate Paths

Once you load the IBMtape device driver with the *alternate_pathing* parameter to be ON, by default, all the available paths for a physical device are enabled. If for some maintenance reason you need to disable a path and do not want to fail over to this path, you can run the following commands:

```
>IBMtapeutil -f /dev/IBMchangerx path (to know what number of this path is)
>IBMtapeutil -f /dev/IBMchangerx disablepath number
      ("number" will be the number of the path you want to disable)
>IBMtapeutil -f /dev/IBMchangerx path
      (to make sure the path is labeled as "disabled")
```

Correspondingly, in the `/proc/scsi/IBMchanger` file, the disabled path will be listed as "Disabled" under the "FO Path" column

To enable a path from a disabled state, you can run the following command:

```
>IBMtapeutil -f /dev/IBMchangerx enablepath number
      ("number" will be the number of the path you want to enable)
```

Linux Device Driver (IBMtape)

Chapter 31. Alternate Pathing Support for Tape Drives

The Linux IBMtape device driver alternate pathing support will configure multiple physical paths to the same device within the device driver and provides the following basic functions:

- Automatic failover to an alternate physical path when a permanent error occurs on one path
- Dynamic load balancing for devices using multiple Host Bus Adapters (HBA)

Automatic Failover

Automatic failover support provides error recovery on an alternate path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) in a Linux host that is connected through a switch to the tape drive. This connection provides two physical paths to same tape drive for redundancy if one path from an HBA to the drive fails.

When the Host Bus Adapter device driver is loaded into the running system, the HBA detects two logical devices of the tape drive. For example, `/dev/IBMtape0` and `/dev/IBMtape1` will be created by the `IBMtaped` daemon or the `IBMtapeconfig` script. Each logical device is a physical path to the same tape drive. A backup and restore application can open and use only one logical device at a time, either `/dev/IBMtape0` or `/dev/IBMtape1`, because they represent the same physical device.

Without the IBMtape alternate pathing support, if an application opens `/dev/IBMtape0` and a permanent path error occurs (because of an HBA or cable failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (`/dev/IBMtape1`), but the application has to be restarted from the beginning. A long backup or restore operation could have been in progress when the path error occurred. Sometimes manual failover can require operator intervention to reset the drive because a SCSI Reservation could still exist on the failing HBA path.

When the alternate pathing support is enabled on both `/dev/IBMtape0` and `/dev/IBMtape1`, the device driver configures them internally as a single device with multiple paths. The application can still open and use only one logical device at a time (either `/dev/IBMtape0` or `/dev/IBMtape1`). If an application opens `/dev/IBMtape0` and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (`/dev/IBMtape1`). If successful, the current operation continues on the alternate path without interrupting the application. The failover error recovery first restores the previous device state, SCSI reservation, and tape position, then retries the failing operation.

Notes:

1. This function only supports the fibre channel 3580 Generation II or above tape drives inside the IBM TotalStorage Ultrium tape libraries and IBM 3592 tape drives. It requires the feature code 1681 of the library to enable the function.
2. If the configuration is changed, reload the Host Bus Adapter device drivers is required

Dynamic Load Balancing

The dynamic load balancing support optimizes resources for devices that have physical connections to multiple Host Bus Adapters (HBA) in the same machine. When an application opens a device that has multiple HBA paths configured, the device driver determines which path has the HBA with the lowest usage, and assigns that path to the application. When another application opens a different device with multiple HBA paths, the device driver again determines the path with the lowest HBA usage and assigns that path to the second application. The device driver will update the usage on the HBA assigned to the application when the device is closed. Dynamic load balancing will use all Host Bus Adapters whenever possible and balance the load between them to optimize the resources in the machine.

For example, consider a machine with two Host Bus Adapters, HBA1 and HBA2, with multiple tape drives attached. Each tape drive is connected to both HBA1 and HBA2. Initially, no tape drives are in use. When the first application opens a tape drive for use, the device driver will assign the application to use HBA1. When a second application opens a tape drive for use, the device driver will assign the second application to use HBA2. A third application would be assigned to HBA1 and a fourth application would be assigned to HBA2. There would be two applications using HBA1 and two applications using HBA2.

If the first application finishes and closes the device, there would now be one application using HBA1 and two applications using HBA2. When the next application opens a tape drive, it would be assigned to HBA1, so again there would be two applications using HBA1 and two applications using HBA2. Likewise, if the second application finishes and closes the device, HBA2 would have one application using it and the next application that opens a tape drive would be assigned to HBA2.

The dynamic load balancing support is independent from the automatic failover support. Regardless of the path assigned initially for load balancing, if that path fails the automatic failover support will attempt recovery on the next available path.

Configuring and Unconfiguring Alternate Pathing Support

Feature code 1681 of the IBM TotalStorage Ultrium tape library is required to support alternate pathing on the 3580 tape drives (only Generation II or above are supported). With this feature code, a data path failover feature key will be provided with the library. The feature key is a string with 16 characters and numbers.

Alternate pathing support is not enabled automatically when the device driver is installed. You will need to provide the data path failover feature key and enable the Linux IBMtape device driver parameter *alternate_pathing*.

To enable failover support in the IBMtape device driver software, perform the following steps after installing the IBMtape *rpm* package:

```
>IBMtaped stop (stop the IBMtaped daemon)
>rmmmod IBMtape (unload the IBMtape driver from the memory)
```

Add the following line in your `/etc/modules.conf` file for 2.4 kernels or `/etc/modprobe.conf.local` file for 2.6 kernels:

```
options IBMtape alternate_pathing=1 dpf_keys="abcdefghijklmnop"
```

"abckdefghijklmnop" is an example of a data path failover feature key. If you have multiple libraries and multiple data path failover feature keys, input your keys as follows:

```
dpf_keys="key1;key2;..."
```

Save the file, then run the following commands:

```
>depmod
>modprobe IBMtape (re-load the IBMtape driver into memory)
>IBMtaped (re-start IBMtaped daemon)
```

You can ignore the "Unresolved symbols in /lib/modules/<your kernel name>/drivers/scsi/IBMtape.o" message after the *depmod* command. You can check if the IBMtape driver has recognized multiple paths for your tape drive by reading the /proc/scsi/IBMtape file:

```
>cat /proc/scsi/IBMtape
```

If your tape drive lists "Primary" or "Alternate" under "FO Path", you have successfully enabled data path failover feature for your tape drive. If "NA" is listed under "FO Path", then the data path failover is not enabled. After the alternate pathing support is enabled, it remains set until the IBMtape driver is reloaded with the *alternate_pathing* driver parameter set to OFF . The alternate pathing setting is retained even if the system is rebooted. To turn off the data path failover feature in the IBMtape device driver, do the following:

```
>IBMtaped stop
>rmmmod IBMtape
```

Delete the following line in your /etc/modules.conf file:

```
options IBMtape alternate_pathing=1
>depmod
>modprobe IBMtape
>IBMtaped
```

Primary and Alternate Paths

When the IBMtape device driver is loaded into kernel memory with alternate pathing support enabled, the first logic device that IBMtape sees always becomes the primary path. The other logical devices that IBMtape sees will be configured as the alternate paths. The device driver supports up to 16 physical paths for a single device. The primary and alternate path information can be obtained by the following command:

```
>cat /proc/scsi/IBMtape
```

The following is an example of a /proc/scsi/IBMtape file:

```
IBMtape version: 1.5.3
IBMtape major number: 253
Attached Tape Devices:
```

Number	Model	SN	HBA	FO Path
0	ULT3580-TD2	1234567	QLogic Fibre Channel 2300	Primary
1	ULT3580-TD2	1234567	QLogic Fibre Channel 2300	Alternate

The labeling of a logical device as either a primary or alternate path is for information only, in order to:

- Be able to identify the actual number of physical devices configured on the system and a specific logical device associated with them. Only one logical

device is labeled as the primary path for each physical device. However, many (multiple) logical devices can be labeled as alternate paths for the same devices.

- Provide information about which logical devices configured on the system have alternate pathing support enabled.

Querying Primary and Alternate Path Configuration

You can display the primary and alternate path configuration for all devices by reading the `/proc/scsi/IBMtape` file, as explained in the above section “Primary and Alternate Paths” on page 159. You can also display the primary and alternate path configuration for any device by running the following command:

```
>IBMtapeutil -f /dev/IBMTapex path (IBMTapex is the logical name of any device)
```

This command shows specific information for the primary path and all alternate paths, such as the logical name of the device, the attached host bus adapter, the channel id, the target id, the logical unit number under the target, the current enabled status, and how many paths are configured for the device.

Disabling and Enabling Primary and Alternate Paths

After you load the IBMtape device driver with the `alternate_pathing` parameter set to ON, by default, all the available paths for a physical device are enabled. If for some maintenance reason you need to disable a path and do not want to fail over to this path, you can run the following commands:

```
>IBMtapeutil -f /dev/IBMTapex path
(to know what number of this path is)
>IBMtapeutil -f /dev/IBMTapex disablepath number
(where "number" is the number of the path you want to disable)
>IBMtapeutil -f /dev/IBMTapex path
(to make sure the path is labeled as "disabled")
```

Correspondingly, in the `/proc/scsi/IBMtape` file, the disabled path will be listed as “Disabled” under the “FO Path” column. To enable a path from a disabled state, you can run the following command:

```
>IBMtapeutil -f /dev/IBMTapex enablepath number
(where "number" is the number of the path you want to enable)
```

Chapter 32. Tape Utility Program (IBMtapeutil)

Installed with the device driver is a tape utility program (*IBMtapeutil*) that exercises or tests the functions of the tape device and the device driver. It also performs basic tape and medium changer operations. The tape utility program can be used in either the interactive or the command-line mode.

The C source code for the *IBMtapeutil* program contains a sample of the interface to the device driver and the *ioctl* commands supported by the device driver. *IBMtapeutil.c* can be found in the tar file *IBMtapeutil.x.x.x.tar* in the Linux directory on your Tape Device Drivers and SMI-S Agent CD.

Interactive Mode

The interactive mode for the tape utility program can be invoked from the command line using the *IBMtapeutil* command. A list of general subcommands, medium changer subcommands, tape subcommands, and service aid subcommands is displayed. You must open a device before using these commands and operations.

To open a device:

1. Select **Open a Device** from General Commands.
2. Enter the name of the device special file. Use any special file that exists for the device, for example, */dev/IBMtape0*, */dev/IBMtape0n*, or */dev/IBMchanger0*.
3. Enter the *Read/Write*, *Read Only*, *Write Only*, or *Append* mode to open a device. These modes apply only to the tape devices. *Append* mode uses *Write Only* for file access permission. After you open a device, select a command using the appropriate number for the command from the window. Some commands require additional information after they are selected from the window. If an error occurs running the command, the error number, the error text, and the device sense data (if applicable) are displayed.

Command-Line Mode

The command-line mode for the tape utility program (*IBMtapeutil*) provides the same basic tape and changer commands as the interactive mode. Invoke the *IBMtapeutil* command from the Linux command line or from within a shell script. If you enter the *IBMtapeutil* command without any arguments, the interactive mode is invoked. The syntax for the command-line interface of the tape utility program is:

```
IBMtapeutil -f Device Subcommand [Subcommand ...]
```

Note:

1. *Device* is the name of the device special file (for example, */dev/IBMtape0*).
2. *Subcommand* is any valid command for the device. Multiple subcommands can be combined in a single command to perform more than one operation. The subcommands are processed one at a time in the order specified on the command line. For help information about the subcommands and their syntax, enter the *IBMtapeutil ?* command on the Linux command line. The following Help information is displayed:

General Subcommands:

tur	inquiry [Page]	print "Text"
reserve	release	reqsense
qryversion	logpage "Page"	modepage "Page"
qrypath	enablepath "primary" number	
path	disablepath "primary" number	

Tape Subcommands:

bsf [Count]	bsr [Count]	eof [Count]
fsf [Count]	fsr [Count]	weof [Count]
fsfm [Count]	bsfm [Count]	asf [Count]
compress	tell	seek [Count]
nocompress	rewind	sync
load	erase	display "Message"
unload	retension	read -d Destination [-c Count]
qrypos	seod	write -s Source
setpos [Blockid]	status	rtest [-b Blocksize] [-c Count] [-r Repetition]
offline	parms	wtest [-b Blocksize] [-c Count] [-r Repetition]
rewoffl	list	rwtest [-b Blocksize] [-c Count] [-r Repetition]
prevent	lock	setblk [Count]
allow	unlock	density
qryinquiry	qrysense	append mtdevice

Medium Changer Subcommands:

allow	prevent	audit [Address [Count]]
inventory	mount [Slot]	position "Destination"
elementinfo	unmount [Slot]	move "Source" "Destination"
devids	exchange "Source" "Dest1" "Dest2"	

Service Aid Subcommands:

dump [Filename]	forcedump	ucode "Name"	resetdrive
-----------------	-----------	--------------	------------

General Subcommands

The following general subcommands are available for the tape and medium changer devices. Items specified with square bracket ([]) delimiters are optional.

disablepath "primary" | number

The disablepath subcommand is used to disable a path to a medium changer. IBM recommends that you leave all paths enabled, but this subcommand can be used to disable a path. If "primary" is specified, the primary path is disabled. If a numeric value is specified, that specific path will be disabled. Use the path command to obtain a list of the paths. If there is only one path available and you attempt to disable it, this subcommand will fail with an illegal argument. If you attempt to disable a data path (to the tape drive), a "function not implemented" error will be returned. This subcommand is only available in IBMtapeutil version 1.2.0 or higher, and the IBMtape device driver, version 1.4.1 or higher, on libraries that have the Control Path Failover feature enabled.

enablepath "primary" | number

The enablepath subcommand is used to enable a path to a medium changer that has been previously disabled with the disablepath subcommand. If "primary" is specified, the primary path is enabled. If a numeric value is specified, that specific path will be enabled. Use the path command to obtain a list of the paths. If you attempt to enable a data path (to a tape drive), a "function not implemented" error

will be returned. This subcommand is only available in IBMtapeutil version 1.2.0 or higher, and the IBMtape device driver, version 1.4.1 or higher, on libraries that have the Control Path Failover feature enabled.

inquiry [Page]

This subcommand issues the SCSI Inquiry command to the device for either standard inquiry data, when no *page* parameter is specified or for a specific inquiry page. The *page* parameter must be specified as a hex value. For example:

```
# Get standard inquiry data
IBMtapeutil -f /dev/IBMtape0 inquiry
# Get inquiry page x'83'
IBMtapeutil -f /dev/IBMtape0 inquiry 83
```

logpage "Page"

This subcommand issues the SCSI Log Sense Page command to the device for the specified page and displays the log sense data. The *page* parameter must be specified as a hex value. For example:

```
# Get log page x'31'
IBMtapeutil -f /dev/IBMtape0 logpage 31
```

modepage "Page"

This subcommand issues the SCSI Mode Sense Page command to the device for the specified page and displays the mode sense data. The *page* parameter must be specified as a hex value. For example:

```
# Get mode page x'1D'
IBMtapeutil -f /dev/IBMtape0 modepage 1d
```

path

This subcommand shows all the available paths to a medium changer. This subcommand is only available in IBMtapeutil version 1.2.0 or higher, and the IBMtape device driver, version 1.4.1 or higher, on libraries that have the Control Path Failover feature enabled.

print "Text"

This subcommand prints the associated text to standard output. It can be used at any time to display the progress of the subcommands. For example:

```
# Rewind, erase current tape and backup myfile.tar
IBMtapeutil -f /dev/IBMtape0 rewind \
erase \
print "Writing myfile.tar" \
write -s myfile.tar
```

qrypath

This subcommand shows the primary path and the first alternate path for a medium changer. This subcommand is only available in IBMtapeutil version 1.2.0 or higher, and the IBMtape device driver, version 1.4.1 or higher, on libraries that have the Control Path Failover feature enabled.

qryversion

This subcommand issues the `QUERY_DRIVER_VERSION ioctl` command. It prints out the current version of the IBMtape driver.

release

This subcommand explicitly releases a device and makes it available for other hosts by issuing the SCSI Release command.

reqsense

This subcommand issues the SCSI Request Sense command and prints out the sense data.

reserve

This subcommand explicitly reserves a device by issuing the SCSI Reserve command.

tur

This subcommand issues the SCSI Test Unit Ready command to the device.

Tape Subcommands

The following tape subcommands are available for the tape devices:

allow

This subcommand issues the SCSI Allow Medium Removal command to the device to allow medium removal by an operator. It is normally used after the *prevent* subcommand to restore the device to the default state.

append

This subcommand opens the device in *Append* mode. The file access permission is set to *Write Only*.

asf [Count]

This subcommand places the tape at the beginning block of *count* files from the beginning of the tape. Positioning is done by first rewinding the tape and then spacing forward over *count* filemarks. The default count is 1.

bsf [Count]

This subcommand backward spaces *count* filemarks. The tape is positioned on the beginning of the last block of the previous file. An optional *count* can be specified. The default is 1.

bsfm [Count]

This subcommand backward spaces *count* filemarks then positions the tape on the end of tape (EOT) side of the filemark. An optional *count* can be specified. The default is 1.

bsr [Count]

This subcommand backward spaces *count* records. An optional *count* can be specified. The default is 1.

compress and nocompress

These subcommands turn hardware compression On or Off.

density

This subcommand issues the SCSI Report Density Support command for all supported media and for the current media loaded in the drive, and displays the results. If the drive is not loaded, the current media density is not reported.

display "Message"

This subcommand is not valid for Ultrium devices.

eof [Count] and weof [Count]

These subcommands write *count* filemarks. An optional *count* can be specified. The default is 1.

erase

This subcommand erases the tape from the current position through the physical end of the tape.

fsf [Count]

This subcommand forward spaces *count* filemarks. The tape is positioned on the first block of the next file. An optional *count* can be specified. The default is 1.

fsfm [Count]

This subcommand forward spaces *count* filemarks, then positions the tape on the beginning of tape (BOT) side of the filemark on the last block of the previous file. An optional *count* can be specified. The default is 1.

fsr [Count]

This subcommand forward spaces *count* records. An optional *count* can be specified. The default is 1.

list

This subcommand rewinds the tape to the beginning of tape and displays the contents. The output lists filemarks and the size of each record found on the tape until the end of data is reached. If a record has more than 256 KB, it returns a read error. When the command completes, the tape is at end of data. The output generated from this subcommand can be large, depending on the amount of data on the tape, and should usually be directed to a file. For example:

```
# List tape contents to file
IBMtapeutil -f /dev/IBMtape0 list > tape.list
```

load

This subcommand issues the SCSI Load command to load a tape.

lock

This subcommand locks the tape drive door.

mtdevice

This subcommand is not valid for IBM Ultrium devices.

offline, rewoffl, and unload

These subcommands rewind and unload the tape.

parms

This subcommand issues the `STIOCQRYP ioctl` command and displays the current tape drive, media, and device driver parameters. The user is then prompted for input to set parameters to new values, and then issues the `STIOCSETP ioctl` command to do so.

prevent

This subcommand issues the SCSI Prevent Medium Removal command to the device to prevent medium removal by an operator until the *allow* subcommand is issued or the device is reset.

qryinquiry

This subcommand issues the `STIOCQRYINQUIRY ioctl` command.

qrypos

This subcommand issues the `STIOCQRYPOS ioctl` command for the current logical and physical tape positions and displays the data. In addition, the current tape position is saved and can be restored using a subsequent *setpos* subcommand. For example:

```
# Append myfile.tar to the end of tape and then read back
IBMtapeutil -f /dev/IBMtape0 append \
qrypos \
write -s myfile.tar \
setpos \
read -d temp.tar
# Verify myfile.tar was written correctly
diff myfile.tar temp.tar
```

qrysense

This subcommand issues the `STIOCQRYSENSE ioctl` command to obtain new sense data.

read -d Destination [-c Count]

This subcommand reads a file, or a specified number of records, from the current position on the tape to the destination file name specified with the *-d* flag. If the optional *count* parameter is used, only the number of records specified with the *-c* flag are read unless a filemark is encountered before the number of specified records. If the *count* parameter is not used, all records up to the next filemark on the tape are read. For example:

```
# Restore myfile.tar from tape
IBMtapeutil -f /dev/IBMtape0 read -d myfile.tar
# Read 3 records from the tape into myfile
IBMtapeutil -f /dev/IBMtape0 read -d myfile -c 3
```

rewind and retension

These subcommands rewind the tape to the beginning of tape (BOT).

rtest [-b Blocksize] [-c Count] [-r Repetition]

This subcommand performs a read test by reading a random data pattern from the tape and verifying that it matches the written data. The *rtest* subcommand can be used after the *wtest* subcommand to verify the data. An optional *block size*, *count*, and *repetition* can be specified with the *-b*, *-c*, and *-r* flags, respectively. If the *block size* is fixed, the *count* specifies the number of blocks to read on each *repetition*. If

the *block size* is zero (variable), the *count* specifies the number of bytes to read on each *repetition*. The default is a *block size* of 10240, a *count* of 20 blocks, and a *repetition* of 1. For example:

```
# R/W test using 256K blocks, 5 megabytes per write, 100 times
IBMtapeutil -f /dev/IBMtape0 rewind \
wtest -b 262144 -c 20 -r 100 \
rewind \
rtest -b 262144 -c 20 -r 100
```

Note: If the number of bytes read is not equal to the number of bytes specified, or there is a data mismatch, an error message is printed out and *errno* is set to 999.

rwtest [-b Blocksize] [-c Count] [-r Repetition]

This subcommand performs a read and write test by writing a random data pattern on the tape, reading it, and verifying that it matches the written data. An optional *block size*, *count*, and *repetition* can be specified with the *-b*, *-c*, and *-r* flags, respectively. If the *block size* is fixed, the *count* specifies the number of blocks to write on each repetition. If the *block size* is zero (variable), the *count* specifies the number of bytes to write on each repetition. The default is a *block size* of 10240, a *count* of 20 blocks, and a *repetition* of 1. For example:

```
# R/W test using 256K blocks, 5 megabytes per write, 10 times
IBMtapeutil -f /dev/IBMtapeutil rwtest -b 262144 -c 20 -r 10
```

Note: If the number of bytes written or read is not equal to the number of bytes specified, or there is a data mismatch while reading, an error message is printed out and *errno* is set to 999.

seek [Count]

If the optional *count* parameter is specified, the tape position is set to the block whose ID is *count*. Otherwise, if the *count* parameter is omitted, the tape position is set to the last position saved using the *tell* subcommand. The *count* can be specified in decimal or in hex (with a leading x).

seed

This subcommand spaces to the end of data on the tape.

setblk [Block Size]

This subcommand sets the block size of the drive to *block size* bytes per record. A *block size* of zero sets the drive to variable block mode. If no *block size* is specified, the drive is set to variable block mode.

setpos [Blockid]

This subcommand issues the SCSI Locate command to the device to set the tape position. If the optional *Blockid* parameter is specified, the tape position is set to the *Blockid*. Otherwise, if the *Blockid* parameter is omitted, the tape position is set to the last position saved using the *grypos* subcommand. The *Blockid* can be specified in decimal or in hex (with a leading x). For example:

```
# Append myfile.tar to the end of tape and then read back
IBMtapeutil -f /dev/IBMtape0n append \
grypos \
write -s myfile.tar \
setpos \
read -d temp.tar
# Verify myfile.tar was written correctly
```

Linux Device Driver (IBMtape)

```
diff myfile.tar temp.tar
# Set tape position to block 32 and leave positioned on close
IBMtapeutil -f /dev/IBMtape0n setpos 32
# Set tape position to block 32 and leave positioned on close
IBMtapeutil -f /dev/IBMtape0n setpos x20
```

status

This subcommand issues the MTIOCGET command and prints out status information about the tape unit.

sync

This subcommand synchronizes or flushes the tape buffers to tape.

tell

This subcommand reports the current block position on the tape and saves the position for use by a subsequent *seek* subcommand.

unlock

This subcommand unlocks the tape drive door.

write -s Source

This subcommand writes the source file specified with the *-s* flag on the tape. For example:

```
# backup myfile.tar to tape
IBMtapeutil -f /dev/IBMtape0 write -s myfile.tar
```

wtest [-b Blocksize] [-c Count] [-r Repetition]

This subcommand performs a write test by writing a random data pattern on the tape. The *rtest* subcommand can be used after the *wtest* subcommand to verify the data that was written. An optional *block size*, *count*, and *repetition* can be specified with the *-b*, *-c*, and *-r* flags, respectively. If the *block size* is fixed, the *count* specifies the number of blocks to write on each *repetition*. If the *block size* is zero (variable), the *count* specifies the number of bytes to write on each *repetition*. The default is a *block size* of 10240, a *count* of 20 blocks, and a *repetition* of 1. For example:

```
# R/W test using 256K blocks, 5 megabytes per write, 100 times
IBMtapeutil -f /dev/IBMtape rewind \
wtest -b 262144 -c 20 -r 100 \
rewind \
rtest -b 262144 -c 20 -r 100
```

Note: If the number of bytes written is not equal to the number of bytes specified, an error message is printed out and *errno* is set to 999.

Medium Changer Subcommands

The following medium changer subcommands are available for the integrated and independent medium changer devices:

allow

This subcommand allows medium removal by an operator. It is normally used after the *prevent* subcommand to restore the device to the default state.

audit [Address [Count]

This subcommand with no parameters issues the SCSI Initialize Element Status command to the device. Using the optional parameters *Address* and *Count* issues the SCSI Initialize Element Status With Range command to the device. The *Address* parameter specifies the starting element address and the *Count* parameter, if used, specifies the number of elements to initialize. If *Count* is omitted, it defaults to 1. For example:

```
# Initialize all elements
IBMtapeutil -f /dev/IBMtape0 audit
# Initialize element 32
IBMtapeutil -f /dev/IBMtape0 audit 32
# Initialize elements 36 to 40
IBMtapeutil -f /dev/IBMtape0 audit 36 5
```

devids

This subcommand issues the SCSI Read Element Status command to the device with the DVCID option for all drive elements and displays the element status information.

elementinfo

This subcommand displays the information returned from the `SMCIOE_ELEMENT_INFO ioctl` command that contains the number of and the first address of each element type.

exchange Source Dest1 Dest2

This subcommand issues the SCSI Exchange Medium command to the device using the *Source*, *Dest1*, and *Dest2* addresses specified. This command performs the equivalent function of two Move Medium commands. The first moves the cartridge from the element address specified by the *Dest1* parameter to the element address specified by the *Dest2* parameter. The second moves the cartridge from the element address specified by the *source* parameter to the element address specified by the *Dest1* parameter. The *Dest2* element address can be the same as *Source* element address. For example:

```
# Exchange cartridge in slot 34 with cartridge in drive address 16
# and return that cartridge to slot 40 (drive must be unloaded first)
IBMtapeutil -f/dev/IBMchanger0 exchange 34 16 40
# Use move medium commands to perform exchange
IBMtapeutil -f/dev/IBMchanger0 move 16 40 move 34 16
```

Note: This command is only supported on the IBM Ultrium 3584 library.

inventory

This subcommand issues the SCSI Read Element Status command for each element type and displays the element status information. No device identifier information is obtained.

mount [Slot]

This subcommand mounts a tape from the specified *slot* address into the drive or from the first full *slot* into the drive if the *slot* is omitted. For example:

```
# Mount cartridge from slot 3
IBMtapeutil -f /dev/IBMchanger0 mount 3
# Mount cartridge from first full slot
IBMtapeutil -f /dev/IBMchanger0 mount
```

Linux Device Driver (IBMtape)

Note: This command is only supported on IBM 3581 Ultrium Tape Autoloader and StorageSmart by IBM Ultrium Tape Autoloader SL7.

move Source Destination

This subcommand issues the SCSI Move Medium command using the *source* and *destination* addresses specified. The element addresses can be obtained using the *elementinfo* subcommand. For example:

```
# Get slot and drive addresses
IBMtapeutil -f /dev/IBMchanger0 elementinfo
# Move cartridge in slot 20 to drive at address 16
IBMtapeutil -f /dev/IBMchanger0 move 20 16
```

position Destination

This subcommand issues the SCSI Position to Element command using the *Destination* specified. For example:

```
# Position to slot at address 20
IBMtapeutil -f /dev/IBMchanger0 position 20
```

prevent

This subcommand prevents medium removal by an operator until the *allow* subcommand is issued or the device is reset.

unmount [Slot]

This subcommand moves a tape from the drive to the specified *slot* address or the first empty one if *slot* is omitted. For example:

```
# Move tape from drive to slot address 4 (tape is already unloaded)
IBMtapeutil -f /dev/IBMchanger0 unmount 4
# Unload tape and move to the first empty slot
IBMtapeutil -f /dev/IBMchanger0 unmount
```

Note: This command is only supported on IBM 3581 Ultrium Tape Autoloader and StorageSmart by IBM Ultrium Tape Autoloader SL7.

Service Aid Subcommands

The following service aid subcommands are available for the devices.

dump [Filename]

This subcommand is implemented using SCSI Pass Through *ioctl*. It stores the dump in the specified *Filename* or, if *Filename* is omitted, in *dump0001.dmp* in the current directory. It should follow a *forcedump* subcommand. For example:

```
# Force a dump and then stores the dump into file /tmp/dump1.dmp
IBMtapeutil -f /dev/IBMtape0 forcedump dump /tmp/dump1.dmp
```

Note: If the device is not supported, or the returned status from SCSI Pass Through is not correct, this operation fails and *errno* is set to 999.

forcedump

This subcommand is implemented using SCSI Pass Through *ioctl*. It forces a dump on a tape device. It is usually followed by the *dump* subcommand.

resetdrive

This subcommand issues a `STIOC_RESET_DRIVE ioctl` command to reset the device.

ucode [Filename]

This subcommand is implemented using SCSI Pass Through *ioctl*. It downloads microcode to the device. *Filename* is a file that contains the *ucode*. For example:

```
# download microcode
IBMtapeutil -f /dev/IBMchanger0 ucode /temp/device.ucode
```

Note: If the device is not supported, or the returned status from SCSI Pass Through is not correct, this operation fails and *errno* is set to 999.

Automatic Cartridge Facility Mode

This parameter is currently not supported for IBM Ultrium devices.

Block Size and SCSI Transfer Size

The minimum and maximum block sizes for the tape device and the maximum SCSI transfer size can be queried using either the interactive mode of the tape utility program and selecting **Query/Set Parameters** under IBMtape Commands or by issuing the *parms* subcommand when in command-line mode.

Configuration Parameters

The configuration parameters can be queried using either the interactive mode of the tape utility program and selecting **Query/Set Parameters** under Tape Commands or the command-line mode by issuing the *parms* subcommand. Some of the parameters can be changed using the interactive mode of the tape utility program and selecting **Query/Set Parameters** under IBMtape Commands.

Note: All configuration parameters are reset to their default values whenever the IBMtape device driver is reinstalled.

Capacity Scaling

This parameter is currently not supported for IBM Ultrium devices.

Logical Write Protect

This parameter is currently not supported for IBM Ultrium devices.

Reserve and Release Commands

When a device is shared by multiple initiators or hosts, the device can be reserved explicitly by a host while application programs are running and released explicitly when the programs are completed. This process ensures that another host cannot use the device until the first host is finished with it. The device driver reserves the device automatically on the *open* call and releases the device on the *close* call. Some backup programs (such as *tar*) can open and close the device multiple times. The reservation must be retained explicitly between the *close* call and the next *open* call.

Linux Device Driver (IBMtape)

A device can be reserved and released explicitly using either the interactive mode of the tape utility program and selecting Reserve Device or Release Device under General Commands, or by issuing the *reserve* and *release* subcommands when in command-line mode. For example:

```
# Reserve device, run tar, and then release device
IBMtapeutil -f /dev/IBMtape0 reserve
tar ... /dev/IBMtape0 ...
IBMtapeutil -f /dev/IBMtape0 release
```

After the *reserve* subcommand is used, the device driver retains the reservation until a *release* subcommand is issued.

Service Aids Commands

The service aids provided with the device driver can be called using the interactive mode of the tape utility program and selecting Service Aids Commands or using the Service Aid Subcommands in the command-line mode of the tape utility program. See Chapter 33, “Tape Drive Service Aids,” on page 173 for details.

Create Special Files

The interactive mode of the tape utility program provides a selection to create special files based on attached devices. It invokes a command *IBMtapeconfig* to remove current */dev/IBMtape* and */dev/IBMchanger* files and generate new ones. The command *IBMtapeconfig* can also be used directly from the Linux command line, assuming this utility program has been installed.

Note: Running *IBMtapeconfig* requires *root* authority.

Chapter 33. Tape Drive Service Aids

The service aids described here are accessible through both the interactive and command-line mode of the *IBMtapeutil*. See "Service Aid Subcommands" in the last chapter ("Tape Utility Program").

Details of Tape Drive Service Aids

The following service aid utilities are available through the utility program *IBMtapeutil* installed with the IBMtape device driver.

- Force Microcode Dump
- Read Dump
- Load Microcode
- Reset Drive

Force Drive Dump

This utility forces a dump operation on the tape drive. After the dump operation is performed, the dump data can be saved to a file using the Read Dump utility. To access this utility, invoke *IBMtapeutil*, then choose "Force Dump" under "Service Aids Commands" in the menu or issue the following command:

```
IBMtapeutil -f /dev/IBMtape0 forcedump
```

Read Dump

This utility transfers the dump data from the device to a file, a diskette, or a tape cartridge. It follows a *force drive dump* operation. To access this utility, invoke *IBMtapeutil*, then choose "Dump Device" under "Service Aids Commands" in the menu or issue the following command:

```
IBMtapeutil -f /dev/IBMtape0 dump [filename]
```

If the filename is not specified, *dump0001.dmp* is used as the default. All the dump files are stored under the current directory.

Load Microcode

This utility downloads microcode to the tape drive or medium changer from a file. To access this utility, invoke *IBMtapeutil*, then choose "Load Ucode" under "Service Aids Commands" in the menu or issue the following command:

```
IBMtapeutil -f /dev/IBMtape0 ucode filename
```

Note: The filename has to be specified. Otherwise, *IBMtapeutil* returns an error.

Reset Drive

This utility resets the tape drive. To access this utility, invoke *IBMtapeutil*, then choose "Reset Drive" under "Service Aids Commands" in the menu or issue the following command:

```
IBMtapeutil -f /dev/IBMtape0 resetdrive
```

Linux Device Driver (IBMtape)

Part 6. Solaris Tape and Medium Changer Device Driver

Chapter 34. Introduction and Product Requirements

This chapter provides an overview of the IBM SCSI Tape and Medium Changer Device Driver for Solaris (IBMtape). It supports a number of IBM Magnetic Tape and Library Subsystem products (see “Hardware Requirements” for specific models):

- IBM TotalStorage Ultrium External Tape Drive 3580
- IBM TotalStorage Ultrium Tape Autoloader 3581
- IBM TotalStorage Ultrium Tape 2U Autoloader 3581
- IBM TotalStorage Ultrium Tape Library 3582
- IBM TotalStorage Ultrium Scalable Tape Library 3583
- IBM TotalStorage UltraScalable Tape Library 3584
- IBM TotalStorage Enterprise Tape System 3592

Purpose

IBMtape provides SCSI and FC attachment for IBM Magnetic Tape and Library Subsystem products to Sun Microsystems SPARC and UltraSPARC platforms running the Solaris operating system, including the Ultra/Enterprise family of servers.

It is designed specifically to take advantage of the features provided by these IBM tape/library subsystems, including full control of the random access medium changer facility (move, element information, and inventory) present in some devices. The goal is to give applications access to the functions required for basic operations, such as backup and restore, and the advanced functions needed by full tape management systems. Whenever possible, IBMtape is designed to take advantage of the IBM tape subsystem features in a manner transparent to the application.

Product Requirements and Compatibility

IBMtape requires and supports the following hardware and software components.

Hardware Requirements

IBMtape requires and supports the following hardware components:

- One or more of the following IBM tape devices:
 - TotalStorage LTO Ultrium Tape Drive 3580
 - TotalStorage LTO Ultrium Tape Autoloader 3581
 - TotalStorage LTO Ultrium Tape 2U Autoloader 3581
 - TotalStorage LTO Ultrium Tape Library 3582
 - TotalStorage LTO Ultrium Scalable Tape Library 3583
 - TotalStorage LTO UltraScalable Tape Library 3584
 - IBM TotalStorage Enterprise Tape System 3592
- One or more of the following FC host bus adapters:
 - QLogic QLA2200F, QLA2310FL, QLA2340, QLA2340L, QLA2342, QLA2342L
PCI Fibre Channel Adapters

Solaris Device Driver (IBMtape)

- Emulex LightPulse LP9802(PCI-X), LP9042DC(PCI-X), LP8000 (PCI), LP9002L (PCI), LP9002C(cPCI), LP8000S (S-Bus), and LP9002S (S-Bus) Fibre Channel Adapters
- JNI FCC/2-6562(PCI), FCE-6460 (PCI) and FCE-1473 (S-Bus) Fibre Channel Adapters
- One or more of the following SCSI High Voltage Differential (HVD) host bus adapters:
 - Sun Microsystems Dual-channel Differential UltraSCSI Host Adapter, PCI (P/N X6541A)
 - Sun Microsystems SBus Ultra Differential Fast/Wide Intelligent SCSI-2 Host Adapter (UDWIS/S) (P/N X1065A)
 - Sun Microsystems Differential Fast/Wide SCSI-2 Host Adapter (DWIS/S) Host Adapter (UDWIS/S) (P/N X1062A)
- One or more of the following SCSI Low Voltage Differential (LVD) host bus adapters on Solaris Version 8, or later:
 - Sun Microsystems PCI Adapter with two fast Ethernet Interfaces and two SCSI Interfaces (P/N X2222A)
 - Sun Microsystems Dual Fast Ethernet + Dual SCSI PCI Adapter (Dual VHDCI)
- The IBM 2108-G07 (IBM SAN Data Gateway) and IBM 2108-R03 (IBM SAN Data Gateway Router) can be used to attach SCSI models of the IBM Ultrium family of products with any of the supported FC-AL host bus adapters

Notes:

1. Using a single Fibre Channel host bus adapter (HBA) for concurrent tape and disk operations is not recommended. Tape and disk devices require incompatible HBA settings for reliable operation and optimal performance characteristics. Under stress conditions (high I/O rates for tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems have been observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and using SAN zoning to minimize contention. IBM is focused on assuring server and storage configuration interoperability. It strongly recommends that your implementation plan includes provisions for separating disk and tape workloads.
2. For IBM Ultrium drives with the Fibre Channel attachment, the Sun Solaris operating system requires that the Fibre Channel addressing mode of the drive be set to hard addressing.

Software Requirements

IBMtape requires and supports the following software components:

- Sun Microsystems Solaris (SPARC) operating system Version 7, 8, or 9
- SCSI Host Bus Adapter (HBA) driver as supplied by either Sun Microsystems or the HBA manufacturer.

See the manufacturer's documentation for the HBA to determine which adapter driver is required.

Software Requirements CPF and DPF Support

To use alternate pathing support, the following feature codes are required for the associated machine type:

Table 15. CPF and DPF Feature Codes

Machine Type	Feature Code
Ultrium 3583	FC 1680
Ultrium 3582	FC 1680
Ultrium 3584	FC 1680
Ultrium 3580	FC 1681

Note: CPF and DPF are not supported on Sun Microsystems Solaris (SPARC) operating system Version 7.

Data Flow

Both data and commands flow between the application program and the tape subsystem through IBMtape. Figure 9 shows the relationships between IBMtape, the application program, the adapter device driver, and the IBM tape subsystem.

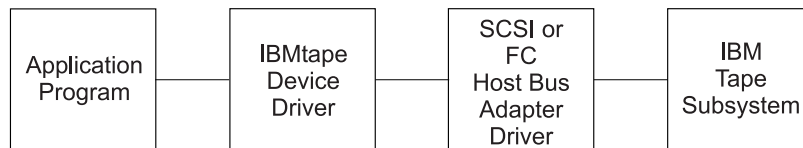


Figure 9. Data Flow

Software Interface to the Device Driver

IBMtape provides the following standard Solaris (UNIX) entry points for IBM tape subsystems:

- Open** This entry point is driven by *open* and *creat* system function calls.
- Write** This entry point is driven by *write* and *writew* system function calls.
- Read** This entry point is driven by *read* and *readv* system function calls.
- Close** This entry point is driven explicitly by the *close* system function call and implicitly by the operating system at program termination.
- ioctl** This entry point is driven by the *ioctl* system function call. It provides a set of tape device, medium changer device, and SCSI specific operations. It allows Solaris applications to access and control the features and attributes of IBM SCSI tape subsystems through the IBMtape Application Programming Interface (API).

Chapter 35. Installation, Removal, and Configuration

IBM SCSI Tape Drive and Medium Changer Device Driver for Solaris is an installable kernel module, supplied as a standard Solaris software package. When installed, its package name is *IBMtape*. The following sections describe installation, removal, configuration, and verification procedures for *IBMtape*. See the Solaris documentation for general information about installable packages.

The *IBMtape* package consists of the device driver and a number of associated files and utilities. For components created during *IBMtape* installation, see Table 16.

Table 16. *IBMtape* Components

Component	Description
<i>/opt/IBMtape</i>	Package subdirectory
<i>/opt/IBMtape/diags_info</i>	Diagnostic script
<i>/opt/IBMtape/tapelist</i>	Utility program
<i>/opt/IBMtape/tmd</i>	Tape Monitor Daemon (TMD) program
<i>/opt/IBMtape/tmd.conf</i>	TMD configuration file, reference version
<i>/etc/tmd.conf</i>	TMD configuration file, working version
<i>/opt/IBMtape/tapeutil</i>	Utility and service aid program
<i>/opt/IBMtape/tapeutil.c</i>	Utility/service program sample source code
<i>/opt/IBMtape/IBMtape.conf</i>	Configuration file, reference version
<i>/usr/kernel/drv/IBMtape</i>	32 bit Kernel device driver module
<i>/usr/kernel/drv/sparcv9/IBMtape</i>	64 bit Kernel device driver module
<i>/usr/kernel/drv/IBMtape.conf</i>	Configuration file, working version
<i>/usr/include/sys/smc.h</i>	Medium changer application programming interface (API) header file
<i>/usr/include/sys/st.h</i>	Tape drive API header file
<i>/usr/include/sys/svc.h</i>	Service aid API header file
<i>/usr/include/sys/oldtape.h</i>	Compatibility API header file

Note: When updating *IBMtape*, the working copy of *IBMtape.conf* located in */usr/kernel/drv* is not overwritten by the package file contents. This allows tape drive configuration options to be preserved across *IBMtape* updates. A reference copy of *IBMtape.conf* is always installed in the */opt/IBMtape* directory.

Examples of installation commands and their results throughout this chapter use a percent sign (%) to indicate the shell prompt.

Preventing Conflicts with Other Device Drivers

IBMtape attempts to claim and operate only the Ultrium devices described in “Hardware Requirements” on page 177. However, the Solaris operating system includes a SCSI tape device driver named *st*, which claims any SCSI compliant tape drive it detects, including devices that *IBMtape* manages. In order to avoid conflicts between *IBMtape* and *st*, you must prevent *st* from claiming and

Solaris Device Driver (IBMtape)

attempting to operate IBMtape owned devices. Likewise, SCSI tape device drivers from other suppliers that are installed on your system must be prevented from claiming IBMtape owned devices.

Attention: Failure to prevent more than one device driver from operating the same SCSI tape drive can cause system panics or data loss on the tape drive.

The following installation and update steps describe how to prevent conflicts between IBMtape and other SCSI tape device drivers.

Installing or Updating IBMtape

Follow these steps to install or update IBMtape. Before starting the step-by-step procedure, note the following general considerations:

- Differential SCSI-2 support must already exist on the system before installing IBMtape. Install and configure one of the supported differential SCSI adapters first, then return to this section. See the HVD SCSI adapter documentation for instructions on installing the adapter and adapter driver.
- You must have root authority to install or remove IBMtape.
- You must reboot the system as part of the IBMtape installation. Take appropriate precautions so this does not adversely affect users or active processes on the system.
- As a consequence of installing or reinstalling IBMtape, device special file numbers under `/dev/rmt` might change. These numbers are assigned by Solaris during the driver attachment process, and the sequencing cannot be specified by the device driver or installer.

Installation Steps

To install IBMtape, follow this procedure:

1. Notify users that system maintenance and a reboot will be performed.
2. Choose a time when all system activity can be stopped to perform the installation.
3. Log on to the target system as root.
4. Ensure that all user and tape drive activity on the system has halted.
5. If tape drives not controlled by IBMtape are installed on the system, list the low density device special files and find the SCSI addresses with which they are associated currently.

```
% ls -l /dev/rmt/*l
lrwxrwxrwx  1 root    root          72 Aug 26 15:47 /dev/rmt/5l ->
../../../../devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,10000/st@2,0:1
```

The last portion of the results shows the controlling device driver and SCSI address. In the preceding example, `/dev/rmt/5l` and the related `5m`, `5h`, and so on, are controlled by the `st` device driver and are associated with the device at SCSI address 2, LUN 0. Record the device type, `/dev/rmt` special file number, owning driver, SCSI target address and LUN; you will need them later during the installation.

For example, suppose an installation has two non-IBM devices owned by `st` at SCSI addresses 2 and 8. The low density devices are accessed as special files `/dev/rmt/5l` and `/dev/rmt/6l`. For the equipment listing after recording the device information, see Table 17 on page 183.

Table 17. IBMtape Install or Update

Device	Old Special File	Old Driver	SCSI Address/LUN (Old)
QIC	/dev/rmt/5l	st	2/0
QIC	/dev/rmt/6l	st	8/0

6. If this is an update to IBMtape, there are already IBMtape owned devices installed. In this case, list the primary device special files and find the SCSI addresses with which they are associated currently.

```
% ls -l /dev/rmt/*st /dev/rmt/*smc
lrwxrwxrwx 1 root other 46 Aug 26 16:36 /dev/rmt/0st ->
../../../../devices/pci@6,4000/scsi@3/IBMtape@b,0:st

lrwxrwxrwx 1 root other 47 Aug 26 16:36 /dev/rmt/1smc ->
../../../../devices/pci@6,4000/scsi@3/IBMtape@b,1:smc
```

The last portion of the results shows the controlling device driver and SCSI address. In the preceding example, /dev/rmt/0st (a SCSI tape drive) is controlled by IBMtape and is associated with the device at SCSI address b, LUN 0. The address is reported in hexadecimal: 0xb=11 decimal. /dev/rmt/1smc (a SCSI medium changer) is associated with the device at SCSI address b, LUN 1. Record the device type, /dev/rmt special file number, owning driver, SCSI target address and LUN; you will need them later during the installation.

For example, suppose an installation has only an IBMtape owned device installed, at SCSI address 8. The device consists of both a tape drive and SCSI medium changer. The tape drive is accessed as /dev/rmt/2st and the medium changer as /dev/rmt/3smc. For a similar equipment listing after recording the device information, see Table 18.

Table 18. Equipment Listing

Device	Old Special File	Old Driver	SCSI Address/LUN (Old)
3580-H11 drive	/dev/rmt/2st	IBMtape	8/0
3581-H17 changer	/dev/rmt/3smc	IBMtape	8/1

7. Choose one of the following methods to prevent conflicts between IBMtape and other SCSI tape device drivers, depending on the equipment attached to your system.

Attention: Failure to prevent more than one device driver from operating the same SCSI tape drive can cause system panics or data loss on the tape drive.

- a. If the system has only IBMtape owned devices attached, follow these steps to prevent st and other non-IBM SCSI tape device drivers from claiming the IBM devices.

- 1) Edit /kernel/drv/st.conf and comment out all SCSI target entries by placing a pound sign (#) in the first column of each target entry. The following example shows the entries for SCSI target addresses 0 and 1 commented out. Repeat this operation for all target entries.

```
#name="st" class="scsi"
#target=0 lun=0;
```

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

- 2) For other non-IBM SCSI tape device drivers that are installed, remove the drivers if they are not needed. If a driver is for SCSI tape devices

Solaris Device Driver (IBMtape)

only, it should not be needed. If a driver is for both tape and disk devices, follow the supplier's instructions to disable its access to all SCSI tape devices.

- b. If the system has a mixture of IBMtape owned devices and other tape drives, follow these steps to configure *st* and other non-IBM SCSI tape device drivers so they control a range of target addresses distinct from the range that IBMtape uses. These steps leave target addresses 7 and 15 unclaimed by all target device drivers, because SCSI adapters typically use one of those two addresses.

- 1) Edit */kernel/drv/st.conf* and comment out SCSI target entries for addresses 7–15 by placing a pound sign (#) in the first column of each target entry. In the following example, the entries for SCSI addresses 7 and 8 have been commented out. Repeat this operation for all entries in the target address range 7–15.

```
#name="st" class="scsi"  
#      target=7 lun=0;
```

```
#name="st" class="scsi"  
# target=8 lun=0;
```

- 2) For other non-IBM SCSI tape device drivers that are installed, follow the suppliers' instructions to disable their access to all SCSI tape devices in the address range 7–15.
- 3) Later, after the *v* package has been installed, you alter its configuration file so it does not use SCSI target addresses in the range 0–7 or address 15.

Now *st* and other non-IBM SCSI tape device drivers have been configured to avoid conflicting with IBMtape.

8. Remove all special file entries under */dev/rmt*. This ensures that stale entries do not exist after the system is rebooted. New entries are created when the system is rebooted.

```
% rm /dev/rmt/*
```

9. If you are updating the level of IBMtape, remove the currently installed *IBMtape* package. If this is a new installation of IBMtape, skip this step.

- a. Use **pkgrm** to remove the current level.

```
% /usr/sbin/pkgrm IBMtape
```

Respond to the *pkgrm* prompts.

- b. Examine the results from *pkgrm*. If you find these messages:

```
...  
Device busy  
Cannot unload module: IBMtape  
Will be unloaded upon reboot.  
...
```

Then one or more IBMtape owned tape drives or tape monitor daemond (TMD) were still in use. Identify the drives and TMD process ID (pid), and end the processes that are using them. If you cannot identify the processes, you must reboot the system to free the tape drive, then continue with the installation from this point.

10. Choose one of the following methods to install the IBMtape package, depending on the package distribution medium and the location of system resources.

Note: If this is a new installation of IBMtape, IBM devices are not yet attached to the system, and you see *pkgadd* error messages similar to the following:

```

...
drvconfig: Driverv) successfully added to system
but failed to attach
## The device driver was unable to detect any supported devices!
## Verify that the device(s) are properly connected and powered on.
## Ensure that the SCSI adapter device driver is installed/configured.
## Then try re-installing the device driver as follows:
## -enter the command: rem_drv IBMtape
## -enter the command: add_drv -m '* 0666 bin bin' IBMtape
## If problems persist, contact your IBM service representative.
pkgadd: ERROR: postinstall script did not complete successfully
...

```

Later, after you have cabled IBM drives to the system and rebooted, the driver is attached normally.

a. If the distribution medium is a CD, follow these steps:

- 1) Determine if volume management is running by displaying the file system type for the */vol* directory. Volume management mounts an NFS filesystem on the */vol* directory. Use the *vol* command to display the file system type.

```

% df -n /vol
/vol      :nfs

```

In this example, the file system type is NFS; therefore volume management is running. If volume management is not running, the */vol* directory is shown as a UFS file system.

- 2) Insert the distribution CD into the local CD-ROM drive.
- 3) If volume management is running, wait for the volume management services to detect and mount the CD-ROM file system. Typically, this completes within 30 seconds.
- 4) If volume management is not running, you must mount the CD-ROM file system manually. To do so, you must know the device special file name for your CD drive. For example, if your CD drive is on controller 1 at SCSI target 6, you would use this command:

```

% mount -o ro /dev/dsk/clt6d0s0 /mnt

```

- 5) Review the contents of the *../Drivers/Solaris/IBMtape.Readme* for any updated instructions.
- 6) Use **pkgadd** to install the driver. Depending on where the CD-ROM file system is mounted, use one of these commands:

```

%/usr/sbin/pkgadd -d /cdrom/cdrom0/Drivers/Solaris/IBMtape.x.x.x.x
%/usr/sbin/pkgadd -d /mnt/Drivers/Solaris/IBMtape.x.x.x.x

```

where *x.x.x.x* is the version number of IBMtape included on the CD.

- 7) If volume management is running, eject the CD:

```

% /usr/bin/eject cd

```
- 8) Otherwise, unmount the device, then eject the media manually:

```

% /usr/bin/unmount /mnt

```

b. If the distribution medium is a package file in a UNIX filesystem, follow these steps. You might have obtained a package file by extracting it from a distribution diskette or by downloading it from IBM's anonymous FTP site. This example presumes a package file named *IBMtape.4.0.2.7*, located in the */tmp* directory.

- 1) If necessary, FTP the package file to the target system. Use binary transfer mode. Place the package file in the target system's */tmp* directory.

Solaris Device Driver (IBMtape)

2) Use **pkgadd** to install the driver:

```
% /usr/sbin/pkgadd -d /tmp/IBMtape.4.0.2.7
```

11. If your system environment includes a mixture of IBMtape owned devices and devices owned by *st* or another third party SCSI tape device driver, you already modified the configuration files for the non-IBM device drivers and restricted them to target addresses in the range 0–6.

Edit *IBMtape.conf*, located in */usr/kernel/drv*, and comment out entries for SCSI target addresses 0–7 and 15 by placing a pound sign (#) in the first column of each line making up the entries. In the following example, the entries for address 0, LUN 0 and address 0, LUN 1 have been commented out. Repeat the operation for all stanzas in the address range 0–7, and address 15. Note that each SCSI target address has a stanza for both LUN 0 and 1.

```
#name="IBMtape" class="scsi"  
# target=0 lun=0  
# block_size=0  
# buffering=1  
# immediate=0  
# trailer=0  
# sili=0;
```

In a SAN environment, the fibre channel HBA driver might map the SCSI target address out of the range 15 and LUN over the number 1. Create a new entry with the mapped SCSI target in *IBMtape.conf*. For example, an IBM tape device is mapped to the SCSI target 32 and LUN 15.

```
name="IBMtape" class="scsi"  
target=32 lun=15  
block_size=0  
buffering=1  
immediate=0  
trailer=0  
sili=0;
```

You can modify the configuration file or use the utility provided by the fibre channel HBA driver to persistently bind the tape device to the expected SCSI target address.

12. Shut down the system. One common method to perform a shutdown is shown here but use your normal procedures.

```
% /usr/sbin/shutdown -y -g0 -i0
```

13. Address or readdress devices as determined by your installation:

- a. If the system has only IBMtape owned devices attached, you can choose addresses in the range 0–6 or 8–14. Leave addresses 7 and 15 unused, because these addresses are used typically by the SCSI adapter.
 - 1) For each device, see the appropriate IBM hardware reference for any special instructions about addressing. Then set the address and record the device type, SCSI address, and LUN. For example, suppose an installation has only IBMtape owned devices attached. An IBM device with tape drive and medium changer is added. It is addressed at target 4, and the information is recorded. For the results, see Table 19.

Table 19. Equipment Listing

Device	Old Special File	Old Driver	SCSI Address/LUN (Old)	SCSI Address/LUN (New)
3580-H11 drive	–	–	–	4/0
3581-H17 changer	–	–	–	4/1

- b. If you are using distinct address ranges to separate tape drives that are IBMtape owned from devices that are owned by *st* or another driver, readdress the tape drives now.
- 1) For each device to be owned by *st* or another SCSI tape device driver, see the manufacturer's hardware reference for any special instructions about readdressing. Then readdress each device to an address in the range 0–6. For each tape drive that is readdressed, record the new SCSI address next to the special file number and old SCSI address that you recorded previously.
 - 2) Readdress all tape drives that are owned by IBMtape to addresses in the range 8–14. See the appropriate IBM hardware references for any special instructions about readdressing. For each tape drive that is readdressed, record the new SCSI address next to the special file number and old SCSI address, if any, that you recorded previously.
- For example, suppose an installation has two non-IBM devices owned by *st* at SCSI addresses 9 and B (12 in decimal). An IBM device with tape drive and medium changer is added. To prevent conflicts between IBMtape and *st*, the non-IBM devices are all placed at addresses in the range 0–6. The new IBM device is addressed in the range 8–14, at address 10 (X'0A'). Depending on the addresses chosen for the non-IBM devices, after readdressing and recording device information, see Table 19 on page 186 for the possible equipment listing.

Table 20. Equipment Listing

Device	Old Special File	Old Driver	SCSI Address/LUN (Old)	SCSI Address/LUN (New)
QIC	<i>/dev/rmt/2l</i>	<i>st</i>	9/0	3/0
QIC	<i>/dev/rmt/3l</i>	<i>st</i>	b/0	5/0
3580-H11 drive	–	–	–	a/0
3581-H17 changer	–	–	–	a/1

Note that the SCSI target address of fibre channel tape device can be over 15.

14. Cable the tape drives to the system, if not yet done. See the manufacturer's hardware references for any special instructions about cabling. Be sure to terminate each SCSI bus properly.
15. Boot the system according to your installation's normal procedures.
16. Log on as root and list the device special files in */dev/rmt* as you did earlier during the installation.

```
% ls -l /dev/rmt/*l
% ls -l /dev/rmt/*st /dev/rmt/*smc
```

Compare the SCSI addresses obtained from *ls* with the readdressed SCSI targets you recorded. Write the new device special file numbers and owning driver next to the matching new SCSI addresses.

For example, suppose an installation had two non-IBM devices owned previously by *st* at SCSI addresses 2 and 8. An IBM device with tape drive and medium changer is added. To prevent conflicts between IBMtape and *st*, the non-IBM devices are all placed at addresses in the range 0–6. The new IBM device is addressed in the range 8–14. Depending on the addresses chosen, after completing installation and recording device information, see Table 21 on page 188 for sample equipment listing entries.

Solaris Device Driver (IBMtape)

Table 21. Sample Equipment Listing

Device	Old Special File	Old Driver	SCSI Address/LUN (Old)	SCSI Address/LUN (New)	New Driver	New Special File
QIC	/dev/rmt/5l	st	2/0	2/0	st	/dev/rmt/0l
QIC	/dev/rmt/6l	st	8/0	0/0	st	/dev/rmt/1l
3580-H11 drive	-	-	-	8/0	IBMtape	/dev/rmt/2st
3581-H17 changer	-	-	-	8/1	IBMtape	/dev/rmt /3smc

Based on the listing, you can see that the tape drive accessed previously as /dev/rmt/5 is now accessed as /dev/rmt/0, the new medium changer is accessible as /dev/rmt/3smc, and so on.

17. Verify operation of the newly installed or readdressed equipment.
18. Notify users of any changed device special files numbers.

Removing IBMtape

Use the **pkgrm** command to remove the IBMtape package from the system.

```
% /usr/sbin/pkgrm IBMtape
```

All active processes using any IBM devices supported by the IBM SCSI Tape and Medium Changer Device Driver for Solaris must be stopped in order for the removal procedure to complete successfully.

Configuration Parameters

When using devices that IBMtape controls, certain device characteristics, such as the default block size, can be controlled through the device driver configuration file. The IBMtape configuration file is named *IBMtape.conf*. The working copy of this file is located in the */usr/kernel/drv* directory.

During installation of IBMtape, the working copy of *IBMtape.conf* is preserved, if it exists. During removal of IBMtape, the working copy of *IBMtape.conf* is not deleted. These conventions allow configuration settings to remain across updates of IBMtape. A reference copy of *IBMtape.conf* with factory default settings is always installed in the */opt/IBMtape* directory.

Note: IBM requires that the Solaris native SCSI tape device driver *st* be configured so that it does not attempt to support SCSI targets that IBMtape controls. See “Preventing Conflicts with Other Device Drivers” on page 181 for more information about multiple driver access to a device.

Attention: Failure to prevent more than one device driver from operating the same SCSI tape drive might cause system panics or data loss on the tape drive.

Configuration settings are applied only at boot time or when IBMtape is unloaded manually from, then reloaded into, memory. If you change configuration settings in *IBMtape.conf*, you can make the changes effective by rebooting the system. As an alternative to rebooting, ensure that no IBMtape owned devices are in use, then issue the following:


```
% /opt/IBMtape/tmd -s          for IBMtape.4.0.9.2 and later
% /usr/sbin/rem_drv IBMtape
% /usr/sbin/add_drv -m '* 0666 bin bin' IBMtape
% /opt/IBMtape/tmd          for IBMtape.4.0.9.2 and later
```

Default settings in *IBMtape.conf* can be overridden for a particular device (and only while the device is kept open) using the *ioctl* application programming interface (API) of the device driver. The parameter settings made through the API revert back to the default values in *IBMtape.conf* the next time that the device is opened. See the *IBM Ultrium Device Drivers: Programming Reference* for more information about changing configuration parameters under program control.

IBMtape.conf contains one stanza for each SCSI target address/LUN pair owned by IBMtape. The reference *IBMtape.conf* file supplied with the package contains a stanza for every possible SCSI target and LUN combination supported by IBM tape subsystems.

The following example shows the stanza for target 0, LUN 0, with IBMtape's default configuration parameter values. The parameter *immediate* is disabled, which means that SCSI commands Write FM, Locate, Load-Unload, Erase, and Rewind complete before returning status:

```
name="IBMtape" class="scsi"
target=0 lun=0
block_size=0
buffering=1
immediate=0
trailer=0
sili=0;
```

The following example shows the stanza for target 0, LUN 0, with IBMtape's default configuration parameter values and the *rewind immediate* mode set on, which causes the SCSI rewind command to return control to the application program before the command actually completes on the tape drive:

```
name="IBMtape" class="scsi"
target=0 lun=0
block_size=0
buffering=1
rew_immediate=1
trailer=0
sili=0;
```

If *immediate* is set to 1 and *rew_immediate* is set to 0, the setting of *rew_immediate* is ignored.

The **name** variable identifies IBMtape as the device driver, and **class** identifies the type of device supported as SCSI.

The **target** and the **lun** variables determine the target address and LUN of IBM devices that are controlled by that stanza. On systems with multiple SCSI adapters, a single target/LUN stanza controls the configuration settings for all devices addressed with that target address and LUN. Thus, two or more supported IBM devices on the system that have the same target and LUN settings but are attached to different SCSI buses are all affected by the configuration parameters of the single stanza having that target address and LUN.

After installation of the IBMtape package is complete, you can eliminate unnecessary probing for devices by commenting out unused target and LUN pairs. In this example, the stanzas for target 0, LUN 0 and target 0, LUN 1 have been

Solaris Device Driver (IBMtape)

commented out. Those address and LUN combinations are not probed, which saves time during a reboot or manual reload of IBMtape. However, if an IBM device is addressed at target 0, LUN 0 or target 0, LUN 1, it is not detected.

```
#name="IBMtape" class="scsi"  
# target=0 lun=0  
# block_size=0  
# buffering=1  
# immediate=0  
# trailer=0  
# sili=0;  
  
#name="IBMtape" class="scsi"  
# target=0 lun=1  
# block_size=0  
# buffering=1  
# immediate=0  
# trailer=0  
# sili=0;
```

The remaining five configuration parameters affect the behavior of the device driver specifically for the IBM device or devices associated with that stanza (target and LUN). All of these parameters are specific only to tape drive device operation and have no effect on medium changer device behavior. The default configuration parameters are adequate for most purposes. However, the values in the configuration file can be modified to suit the specific requirements of the application or the user.

Remember that modifying a value in the configuration file determines the value of the parameter at device open time. While open, the value of a parameter can be altered using an *ioctl* function call, but the change is effective only while the device remains open. Working configuration parameters revert back to the default values (established by the configuration file) when the device is closed and reopened. See the *IBM Ultrium Device Drivers: Programming Reference* for more information about changing configuration parameters by way of program control.

The following list describes the set of configuration parameters recognized by the IBMtape device driver:

- **block_size** (0=variable length) This option specifies the *device block size* that is established with the *SCSI Mode Select* command during an open function call. Until this value is changed, it is the working block size. Variable block size is established using a value of zero. Any other positive value represents a fixed block size. The maximum supported block size varies for each tape device. See the appropriate hardware reference manual.

Note: IBMtape does not allow odd byte count fixed block *reads* or *writes*. For instance, a fixed block size of 4096 or 4098 is allowed, but 4097 is not. If you attempt to read or write using an odd byte-count fixed block size, the read or write returns -1, with *errno* set to 22, invalid argument. If you must read or write odd byte count blocks, set block size to 0 (variable block size), then transfer one block's worth of data per read or write.

- **buffering** (0=Off, 1=On) When a write command is processed, the data is either stored directly on the physical tape or buffered in device hardware. Buffering can be turned On and Off with this option. If buffering is disabled, the effective performance of the device could be seriously degraded, because the tape devices cannot take advantage of their buffering optimization. Buffer flushing (or committing data to the tape) can be controlled by the application through the *STIOC_SYNC_BUFFER ioctl* function.

- **immediate** (0=Off, 1=On) If **immediate** is set to 0, the SCSI commands Write FM, Locate, Load-Unload, Erase, and Rewind return with status when the command actually completes on the tape drive. If **immediate** is set to 1, these commands return with status before the command actually completes.
- **rew_immediate** (0=Off, 1=On) If **rew_immediate** is set to 0, the SCSI Rewind command returns with status when the command actually completes on the tape drive. If it is set to set to 1, the Rewind command returns with status before the command actually completes. If **immediate** is set to 1, the setting of **rew_immediate** is ignored.
- **trailer** (0=Off, 1=On) If a tape drive encounters logical end-of-tape (EOT) during a *write* operation, it returns a check condition status. The driver returns 0 bytes written to notify the application of this EOT situation. A check condition is returned by the tape drive for every subsequent *write* operation when past EOT. If **trailer** is enabled, writing records past EOT is allowed by the device driver. Following the first time the *write* operation notifies the application of EOT, all subsequent EOT notifications are suppressed by the driver, and the actual number of bytes written is returned. When the physical end of media is reached, all *write* operations fail with a return code of -1, regardless of the trailer setting. When trailer is enabled, managing the media past EOT is the application's responsibility.
- **sili** (0=Off, 1=On) Normally, during a *read* operation, if a larger block of data is requested than is actually read from the tape, the tape device raises a check condition. The IBMtape device driver must perform error handling procedures, which adds overhead to the *read* operation. The IBMtape driver does not surface this as an error condition to the application and ultimately returns the actual number of bytes read. However, this driver error processing does result in less than optimum read performance in some scenarios. When SILI mode is enabled, the tape device is forced to Suppress Illegal Length Indication during *read* operations. This eliminates the error processing performed by the driver and results in improved read performance for some scenarios. The actual number of bytes read is still returned to the application in SILI mode.

Adding or Removing Devices

To add support for a new IBM tape subsystem to the system or to remove support for a previously attached IBM tape subsystem, the following steps should be performed:

1. Edit the *IBMtape.conf* file in the */usr/kernel/drv* directory to reflect the change in IBM device support. Either add a new stanza to provide support for a device that is to be added, or remove (comment out) a stanza for a device that is no longer to be supported.
2. When adding support for a new device, ensure that the target and LUN values in the configuration file stanza match the target and LUN settings of the IBM device. See “Configuration Parameters” on page 188 for more information about the *IBMtape.conf* configuration file.
3. Shut down and power Off the host system.
4. Plug the new device into the SCSI bus, or unplug the existing device from the bus. Pay particular attention to proper SCSI cabling and termination.
5. Power On and boot the host system.

Notes:

- a. It is possible to reinitialize the IBMtape device driver without rebooting the system. This is done by first unloading the device driver, then reloading the device driver into kernel memory.

Solaris Device Driver (IBMtape)

- b. For the version of IBMtape.4.0.9.2 and later, the TMD daemon has to be stopped by running the `/opt/IBMtape/tmd -s` command to unload the IBMtape driver from the kernel. Running the `/opt/IBMtape/tmd` command restarts the daemon afterwards to reload the device driver.

The commands to unload the device driver are:

```
% /opt/IBMtape/tmd -s          for IBMtape.4.0.9.2 and later
% /usr/sbin/rem_drv IBMtape
```

The commands to reload the device driver are:

```
% /usr/sbin/add_drv -m '* 0666 bin bin' IBMtape
% /opt/IBMtape/tmd          for IBMtape.4.0.9.2 and later
```

When the IBMtape device driver is reloaded, it reads the *IBMtape.conf* file, and changes made in the file are acknowledged by the device driver. This method can be used to modify configuration parameters.

Note: It is strongly suggested that you power Off the host system and all devices attached to the SCSI bus before adding or removing devices from the SCSI bus. Hot plugging SCSI devices can cause hardware damage and disruption of reliable system operation.

Chapter 36. Special Files

After the IBMtape driver is installed, a set of special files is available for performing I/O operations to each supported device. The device special file names created by the IBMtape device driver are similar to the SCSI tape special files generally used on Solaris systems.

Each tape instance has a set of minor numbers that provides access to the same physical device, but each minor number provides a different function or behavior for the tape subsystem. These minor numbers are accessed through variations of the special file name for that device. The special files are created in the `/dev/rmt` directory. These special files are actually symbolic links to files created within the `/devices` subdirectory hierarchy.

Issuing the `ls -la /dev/rmt` command presents some useful information about these device special files. The following is a representative example of the entries returned by this command for a single IBM tape subsystem. This listing is system dependent; therefore entries varies slightly in format, depending on the platform and the SCSI adapter support. There might also be entries included for other devices that are not supported by the IBMtape device driver.

```
lrwxrwxrwx root other 79 Aug 26 18:54 0smc ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:smc
lrwxrwxrwx root other 78 Aug 26 18:54 0st ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:st
lrwxrwxrwx root other 79 Aug 26 18:54 0stb ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stb
lrwxrwxrwx root other 80 Aug 26 18:54 0stbn ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stbn
lrwxrwxrwx root other 79 Aug 26 18:54 0stc ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stc
lrwxrwxrwx root other 80 Aug 26 18:54 0stcb ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stcb
lrwxrwxrwx root other 81 Aug 26 18:54 0stcbn ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stcbn
lrwxrwxrwx root other 80 Aug 26 18:54 0stcn ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stcn
lrwxrwxrwx root other 79 Aug 26 18:54 0stn ->
  /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,
  10000/IBMtape@2,0:stn
```

These entries show the device hierarchy established to support I/O for an IBM SCSI tape subsystem. The attachment path of the device special files spans from the system board, through the S-bus, to the Sun F/W SCSI adapter (supported by the QLGC, isp@ SCSI adapter device driver), to the IBM device at SCSI target 2 and LUN 0 (supported by the IBMtape device driver). All nine of these special files are associated with the **same** IBM device (device number 0).

Certain device behaviors are determined by which special file in the set is opened for device access. The `smc` special file controls only the medium changer portion of

Solaris Device Driver (IBMtape)

the device and accepts only medium changer operations through the *ioctl* entry point. The *smc* special file does not support the read and write entry points. Only one *st* type special file for a particular device can be opened at any one time. The *smc* special file can be opened concurrently with any one of the *st* special files.

The IBMtape device driver decides which types of special files to create during installation, based on the IBM device type being configured. For the IBM 3580-H11 Ultrium Tape Drive, only the eight *st* special files are created. For the IBM 3581 Ultrium Tape Autoloader, IBM 3583 Ultrium Scalable Tape Library, and IBM 3584 UltraScalable Tape Library, all nine special files shown above are created.

With the information from the previous command, issuing the *ls -la /devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@3,10000* command presents further information about the same special files, as shown in the following example. Again, the actual path information specified in the command varies from system to system.

```
crw-rw-rw- 1 bin bin 109,1696 Aug 26 18:54 IBMtape@2,0:smc
crw-rw-rw- 1 bin bin 109,1664 Aug 26 18:56 IBMtape@2,0:st
crw-rw-rw- 1 bin bin 109,1728 Aug 26 18:54 IBMtape@2,0:stb
crw-rw-rw- 1 bin bin 109,1732 Aug 26 18:54 IBMtape@2,0:stbn
crw-rw-rw- 1 bin bin 109,1688 Aug 26 18:54 IBMtape@2,0:stc
crw-rw-rw- 1 bin bin 109,1752 Aug 26 18:54 IBMtape@2,0:stcb
crw-rw-rw- 1 bin bin 109,1756 Aug 26 18:54 IBMtape@2,0:stcbn
crw-rw-rw- 1 bin bin 109,1692 Aug 26 18:54 IBMtape@2,0:stcn
crw-rw-rw- 1 bin bin 109,1668 Aug 26 18:54 IBMtape@2,0:stn
```

These entries show the *major* and *minor* numbers associated with each special file. Here, the major number is 109 and identifies to the system that the IBMtape device driver is in support of these special files. Major numbers are assigned by the system at the time the driver is installed and varies from system to system. The nine different minor numbers are specific to the special file names and are used by the device driver to determine which special file was used to access the device and control the device behavior accordingly. For example, the minor number 1696 indicates to IBMtape that the device was opened through the *smc* special file. For more information on device special files and major/minor numbers, consult the Solaris *mtio* man pages.

Table 22 shows the special file-naming convention and the associated device attributes recognized by the IBMtape device driver.

Table 22. IBM SCSI Tape/Medium Changer Special Files for Solaris

Special File Name	BSD Compatibility	Rewind on Close	Compression
/dev/rmt/[0-255]smc	N/A	N/A	N/A
/dev/rmt/[0-255]stn	No	No	No
/dev/rmt/[0-255]stcn	No	No	Yes
/dev/rmt/[0-255]st	No	Yes	No
/dev/rmt/[0-255]stc	No	Yes	Yes
/dev/rmt/[0-255]stbn	Yes	No	No
/dev/rmt/[0-255]stcbn	Yes	No	Yes
/dev/rmt/[0-255]stb	Yes	Yes	No
/dev/rmt/[0-255]stcb	Yes	Yes	Yes

Notes:

1. The **BSD (b)** device special file modifies close behavior for non-rewind devices. If the device is opened for **no rewind on close**, in **non-BSD** mode, if the last command before closing the device was a read, then the tape is positioned after the file mark immediately following the last block read. If the device is opened for **no rewind on close**, in **BSD** mode, if the last command before closing the device was a read, the tape is left positioned exactly where it was following the last block read. If the device is opened for **rewind on close**, the **BSD** mode is not relevant.

2. The **no rewind on close (n)** device special file does not rewind the tape during a close operation. Otherwise, the tape is rewound when the device is closed. If the last operation before closing the device was a write or write filemark, then a sufficient number of filemarks is written so that two filemarks follow the data.

For the non-rewind special files, the tapes are positioned between the trailing filemarks before closing. If the device is then reopened and more data is written, it is separated by a single file mark from the previous data.

3. The **compression (c)** device special file determines whether the tape device uses built-in hardware compression while storing data on the tape. The compression mode of the device can also be set to the desired state programmatically through the `STIOC_SET_PARM` ioctl, regardless of the default compression mode established by the special file used originally to open the device.

4. The **smc** special file is created only for IBM tape subsystems that provide medium changer capability. For the IBM 3580 Ultrium Tape Drive, no *smc* special file is created.

5. Only one *st* special file can be opened at any one time. The *smc* special file can be opened by itself or in conjunction with one of the *st* type files. The *smc* special file accepts only medium changer commands. Tape drive commands issued to the medium changer fail with `errno` set to 22, invalid argument.

Aside from the normal configuration with the medium changer answering as a distinct target and LUN pair, some supported devices can be configured with a nonstandard integrated medium changer reporting at the same target and LUN as the tape drive. In this case, both *st* and *smc* special files accept a limited subset of medium changer commands. If you want to use this nonstandard mode, consult the appropriate hardware reference to determine if the drive supports such a configuration.

Solaris Device Driver (IBMtape)

Chapter 37. Alternate Pathing Support for Library Control Path

Note: The library control path failover feature code must be installed prior to enabling the alternate pathing support in the Solaris IBMtape device driver. See “Software Requirements CPF and DPF Support” on page 178 to determine which feature code is required for your machine type.

The Solaris IBMtape device driver alternate pathing support for library control path is also called as Control Path Failover (CPF) support, and will configure multiple physical control paths to the same logical library within the device driver and provide automatic failover to an alternate control path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multi-path connection that consists of two Host Bus Adapters (HBA) that are connected to a library with two drive control ports enabled. The first HBA is connected to the first control port drive, and the second HBA is connected to the second control port drive. This connection provides two physical control paths to the same physical library for redundancy if one path from an HBA to the library fails.

When the tape library is attached to the two HBAs on the Solaris system, each HBA detects a control port to the same library, and two logical devices will be configured with IBMtape driver (for example, `/dev/rmt/0smc` and `1smc`). Each logical device is a physical path to the same library. An application can open and use only one logical device at a time, either `smc0` or `smc1`, because these represent the same physical device.

Without the IBMtape alternate pathing support, if an application opens `0smc` and a permanent path error occurs (because of an HBA, cable, or drive control port failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (`1smc`), but the failing transaction has to be restarted from the beginning.

When the alternate pathing support is enabled, the device driver configures them internally as a single device with multiple paths. The application can still open and use only one logical device at a time (either `0smc` or `1smc`). If an application opens `0smc` and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (`1smc`). If successful, the current operation continues on the alternate path without interrupting the application.

Configuring and Deconfiguring Alternate Pathing Support

Alternate pathing support is enabled automatically by default when the IBMtape device driver is installed on Solaris system. The Solaris IBMtape device driver provides a driver configuration parameter failover for you to enable or disable the library control path failover (CPF) support.

To enable the CPF support for all of the paths, no any action is required.

To disable the CPF support for all of the paths or a particular path, follow the steps below:

Solaris Device Driver (IBMtape)

1. To disable CPF support for all the paths, add and set the failover parameter to off at the beginning of IBMtape.conf file in the directory of /usr/kernel/drv.
2. To disable a particular path, add and set the failover parameter to off in the path entry in *IBMtape.conf* file. For example,
name="IBMtape" class="scsi" target=3 lun=1 failover=0;
3. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:
/opt/IBMtape/tmd -s
/usr/sbin/rem_drv IBMtape
4. Reload the IBMtape driver module in the kernel and start the daemon:
/usr/sbin/add_drv -m ' 0666 bin bin' IBMtape
#/opt/IBMtape/tmd

Primary and Alternate Paths

When the device driver configures a logical device with alternate pathing support enabled, the first device configured always becomes the primary path. When a second or more logical device is configured with alternate pathing support enabled for the same physical device, it configures as an alternate path. The device driver supports up to 16 physical paths for single a device.

The primary and alternate path information can be obtained in the field of "Path Type" running the /opt/IBMtape/tapelist command. For instance,

```
#tapelist -l
Inst# Special File Device Serial No TGT/LUN Ucode WNN WPN Device Physical Path Path Type
-----
686 /dev/rmt/12smc 03584L32 0000000T0039 1/1 402j N/A N/A /devices/pci@If,2000/QLGC,q1a@1/IBMtape@1,1 Primary
688 /dev/rmt/14smc 03584L32 0000000T0039 2/1 402j N/A N/A /devices/pci@If,2000/QLGC,q1a@1/IBMtape@2,1 Alt_path_1
694 /dev/rmt/26smc 03584L32 0000000T0039 5/1 402j N/A N/A /devices/pci@If,2000/QLGC,q1a@1/IBMtape@5,1 Alt_path_2
```

The labeling of a logical device as either a primary or alternate path is for information only; it is used to:

1. Identify the actual number of physical devices configured on the system and a specific logical device associated with them. There will be only one logical device labeled the primary path for each physical device. However, there can be multiple logical devices labeled as an alternate path for the same devices.
2. Provide information about which logical devices configured on the system have alternate pathing support enabled.

Querying Primary and Alternate Path Configuration

You can display the primary and alternate path configuration for all devices with the tapelist utility.

You can also display the primary and alternate path configuration for any device running the tapeutil utility and selecting the option 54 "Query Device Paths". For example,

```
Path list in the device:
    Path 0 Path 1 Path 2

Current active path: Path 0

Enter path number or <enter> for all of the paths:

Instance Number ..... 201
SCSI Target ..... 4
```

```

SCSI LUN ..... 1
Serial Number ..... 000001301084
Ucode Level ..... 227B
Path World Wide NN ..... N/A
Path World Wide PN ..... N/A
Path Type ..... Primary
Path ..... Enable
Device Physical Path ..... /devices/pci@1f,2000/scsi@1/IBMtape@4,1

Instance Number ..... 231
SCSI Target ..... 3
SCSI LUN ..... 1
Serial Number ..... 000001301084
Ucode Level ..... 227B
Path World Wide NN ..... N/A
Path World Wide PN ..... N/A
Path Type ..... Alt_path_1
Path ..... Enable
Device Physical Path ..... /devices/pci@4,2000/pci@1/fibre-channel@4/IBMtape@3,1

Instance Number ..... 293
SCSI Target ..... 2
SCSI LUN ..... 1
Serial Number ..... 000001301084
Ucode Level ..... 227B
Path World Wide NN ..... N/A
Path World Wide PN ..... N/A
Path Type ..... Alt_path_2
Path ..... Enable
Device Physical Path ..... /devices/pci@6,4000/lpfc@4/IBMtape@2,1

```

This option shows specific information for each path, such as the instance number for the logical device, device serial number, the target id, the logical unit number, the current enabled status, path type and so on. It also indicates how many paths are configured for the device, and which path is active currently.

Disable and Enable Primary and Alternate Paths

Once you install the IBMtape device driver, by default, all the available paths for a physical device are enabled. If for some maintenance reason you need to disable a path and do not want to fail over to this path, you can run the following steps:

1. Turn the failover parameter to off in the path entry in the *IBMtape.conf* file.

For example,

```
name="IBMtape" class="scsi" target=3 lun=1 failover=0;
```

2. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

```
# /opt/IBMtape/tmd -s
# /usr/sbin/rem_drv IBMtape
```

3. Reload the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m ' 0666 bin bin' IBMtape
# /opt/IBMtape/tmd
```

To enable a path from a disabled state, you can run the following steps:

1. Remove the parameter of failover=0 in the path entry in *IBMtape.conf* file. For example,

```
name="IBMtape" class="scsi" target=3 lun=1"
```

2. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

Solaris Device Driver (IBMtape)

```
# /opt/IBMtape/tmd -s  
# /usr/sbin/rem_drv IBMtape
```

3. Reload the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m ' 0666 bin bin' IBMtape  
# /opt/IBMtape/tmd
```

Chapter 38. Alternate Pathing Support for Tape Drives

Note: The tape drive failover feature code must be required prior to enabling the alternate pathing support in the Solaris IBMtape device driver. Refer to “Software Requirements CPF and DPF Support” on page 178 to determine which feature code is required for your machine type.

The IBMtape device driver alternate pathing support for tape drives is also called Data Path Failover (DPF) support. It enables you to configure multiple physical paths to the same device within the device driver, and provides automatic failover to an alternate physical path when a permanent error occurs on one path.

Automatic Failure

The automatic failover support provides error recovery on an alternate path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) in a Solaris host that is connected through a switch to the tape drive. This connection provides two physical paths to the same tape drive for redundancy, if one path from an HBA to the drive fails.

When the drive is attached on the Solaris system, each HBA detects a tape drive, and two logical devices will be configured (for example, 0st and 1st). Each logical device is a physical path to the same tape drive. A backup and restore application can open and use only one logical device at a time, either 0st or 1st, because they represent the same physical device.

Without the IBMtape alternate pathing support, if an application opens 0st and a permanent path error occurs (because of an HBA or cable failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (1st), but the application has to be restarted from the beginning. A long backup or restore operation could have been in progress when the path error occurred. Sometimes manual failover can require operator intervention to reset the drive because a SCSI Reservation could still exist on the failing HBA path.

When the alternate pathing support is enabled on both 0st and 1st, the device driver configures them internally as a single device with multiple paths. The application can still open and use only one logical device at a time (either 0st or 1st). If an application opens 0st and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path (1st). If successful, the current operation continues on the alternate path without interrupting the application. The failover error recovery first restores the previous device state, SCSI Reservation, and tape position, then retries the failing operation.

Notes:

1. This function is not supported on IBM Ultrium 3580 Gen I Fibre Channel tape drive, nor is it supported for devices that are attached through an IBM San Data Gateway
2. When devices applied to the DPF support are initially configured at a specific SCSI ID and physical connection (drive port, host bus adapter, and switch

number/port, if applicable) and in the available state, changing the physical device address/connection without either rebooting or deconfiguring and reconfiguring the devices produces unpredictable results and is not supported.

Configuring and Deconfiguring Alternate Pathing Support

Alternate pathing support for the tape drives is not enabled automatically when the device driver is installed. It must be configured initially on each logical device after installation. When alternate pathing support is enabled for a logical device, it remains set until the device is deleted or the support is deconfigured. The alternate pathing setting is retained even if the system is rebooted.

Alternate pathing support can be enabled on all configured devices at one time, or it can be enabled or disabled selectively by logical device. It might be desirable at times to configure some, but not all, logical paths to a device with the support enabled. Follow the above steps to enable the DPF support:

1. Enter the DPF feature key(s) in the parameter `dpf_keys` at the beginning of the *IBMtape.conf* file in the directory of `/usr/kernel/drv`, such as

```
dpf_keys="A729E60F7B119411, C7A0B9ef2c1a4360, a729e60f7b118460";
```

Note: The parameter `dpf_keys` is in the format "key1, key2, key3, ...". Each key is 16 characters long with a separation "," and a space ". IBMtape driver supports up to 36 dpf keys.

2. To enable the support globally on all currently configured devices, add an entry of the `dpf_support` parameter and turn it on at the beginning of the *IBMtape.conf* file, such as

```
dpf_support=1;
```

3. Or, to enable a particular path, add the flag `dpf_support=1` in the path entry in *IBMtape.conf* file. For example,

```
name="IBMtape" class="scsi" target=3 lun=0 dpf_support=1;
```

4. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

```
# /opt/IBMtape/tmd -s  
# /usr/sbin/rem_drv IBMtape
```

5. Reload the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m ' 0666 bin bin' IBMtape  
# /opt/IBMtape/tmd
```

This will deconfigure all devices to remove or comment out the `dpf_support` parameter in the *IBMtape.conf* file, and reboot the system or deconfigure and reconfigure all devices. For example,

1. To disable the support globally on all currently configured devices, remove or comment out the entry `dpf_support=1` at the beginning of the *IBMtape.conf* file.

```
# dpf_support=1;      * comment out the parameter
```

2. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

```
# /opt/IBMtape/tmd -s  
# /usr/sbin/rem_drv IBMtape
```

3. Re-load the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m ' 0666 bin bin' IBMtape  
# /opt/IBMtape/tmd
```

To disable the support on a single logical device, follow these steps:

1. Enter the DPF feature key(s) in the parameter `dpf_keys` at the beginning of the *IBMtape.conf* file in the directory of `/usr/kernel/drv`.
2. To enable the support globally on all currently configured devices, add an entry of `dpf_support=1` at the beginning of the *IBMtape.conf* file, such as

```
dpf_support=1;
```
3. To disable a particular path, add the parameter `dpf_support` and turn it off in the path entry in the *IBMtape.conf* file. For example,

```
name="IBMtape" class="scsi" target=3 lun=0 dpf_support=0;
```
4. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

```
# /opt/IBMtape/tmd -s
# /usr/sbin/rem_drv IBMtape
```
5. Reload the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m ' 0666 bin bin' IBMtape
# /opt/IBMtape/tmd
```

Primary and Alternate Paths

When the device driver configures a logical device with alternate pathing support enabled, the first device configured always becomes the primary path. When a second logical device is configured with alternate pathing support enabled for the same physical device, it configures as an alternate path. A third logical device is configured as the next alternate path, and so on. The device driver supports up to 16 physical paths for a single device.

For example, if 2st is configured first, then 19st, the `/opt/IBMtape/tapelist` command output will be similar to the following:

```
#tapelist -l
Inst# Special File Device      Serial No TGT/LUN  Ucode WWNW WWPW Device Physical Path  Path Type
-----
691 /dev/rmt/2st  ULT3580-TD2  1110000877  4/0 45DH  500308C13D25C003  N/A /devices/pci@If,2000/QLGC,
qla01/IBMtape04,0 Primary
668 /dev/rmt/19st  ULT3580-TD2  1110000877  2/0 45DH  500308V13D25C003  N/A /devices/pci@If,4000/JNI,
FCR02/IBMtape02,0 Alt_path_1
```

The labeling of a logical device as either a primary or alternate path is for information only, in order to:

1. Be able to identify the actual number of physical devices configured on the system and a specific logical device associated with them. There will be only one logical device labeled the primary path for each physical device. However, there can be many (multiple) logical devices labeled as an alternate path for the same devices.
2. Provide information about which logical devices configured on the system have alternate pathing support enabled.

Querying Primary and Alternate Path Configuration

You can display the primary and alternate path configuration for all devices with the `tapelist` utility.

You can also display the primary and alternate path configuration for any device running the `tapeutil` utility and selecting option 54 **Query Device Paths**. For example,

Solaris Device Driver (IBMtape)

```
Path list in the device:
  Path 0 Path 1
Current active path: Path 0
```

Enter path number or <enter> for all of the paths:

```
Instance Number ..... 691
SCSI Target ..... 4
SCSI LUN ..... 0
Serial Number ..... 1110000877
Ucode Level ..... 45DH
Path World Wide NN ..... 500308C13D25C003
Path World Wide PN ..... N/A
Path Type ..... Primary
Path ..... Enable
Device Physical Path ..... /devices/pci@1f,2000/QLGC,q1a@1/IBMtape@4,0

Instance Number ..... 668
SCSI Target ..... 2
SCSI LUN ..... 0
Serial Number ..... 1110000877
Ucode Level ..... 45DH
Path World Wide NN ..... 500308C13D25C003
Path World Wide PN ..... N/A
Path Type ..... Alt_path_1
Path ..... Enable
Device Physical Path ..... /devices/pci@1f,4000/JNI,FCR@2/IBMtape@2,0
```

This option shows specific information for each path, such as the instance number for the logical device, device serial number, the target id, the logical unit number, the current enabled status, the path type, and so on. It also indicates how many paths are configured for the device, and which path is the current active path.

Disable and Enable Primary and Alternate Paths

Once you enter the dpf feature key(s), turn on the dpf_support parameter in the *IBMtape.conf* file, and install the IBMtape device driver, all the available paths for a physical device are enabled. If for some maintenance reason you need to disable a path and do not want to fail over to this path, you can run the following steps:

1. Add the parameter dpf_support=0 in the path entry in the *IBMtape.conf* file. For example,

```
name="IBMtape" class="scsi" target=4 lun=0 dpf_support=0;
```

2. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

```
# /opt/IBMtape/tmd -s
# /usr/sbin/rem_drv IBMtape
```

3. Re-load the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m ' 0666 bin bin' IBMtape
# /opt/IBMtape/tmd
```

To enable a path from a disabled state, you can run the following steps:

1. Remove the parameter of dpf_support=0 at the path entry in the *IBMtape.conf* file. For example,

```
name="IBMtape" class="scsi" target=4 lun=0;
```

2. Stop the TMD (tape monitor daemon) running on the system and unload the IBMtape driver module from the current kernel:

```
# /opt/IBMtape/tmd -s
# /usr/sbin/rem_drv IBMtape
```


3. Reload the IBMtape driver module in the kernel and start the daemon:

```
# /usr/sbin/add_drv -m '0666 bin bin' IBMtape  
# /opt/IBMtape/tmd
```

Solaris Device Driver (IBMtape)

Chapter 39. Service and Diagnostic Aids

The following section describes the service and diagnostic aids that are part of the IBMtape package. It discusses the procedure for verifying that the device driver was installed correctly, provides basic problem determination guidelines, and outlines the utility program included with the IBMtape package.

Functional Verification

To verify that the installation of the IBMtape package was successful, enter the following command:

```
/usr/bin/pkginfo IBMtape
```

The following information should be displayed:

```
system  IBMtape  IBM SCSI Tape & Medium Changer Device Driver x.x.x.x
```

where **x.x.x.x** is the version of the device driver.

To verify that device driver support for a specific IBM tape subsystem attached to the system is functioning correctly, enter the following command:

```
/opt/IBMtape/tapeutil -f /dev/rmt/nst -o chk -v
```

substituting for **n** the number associated with the device special file assigned to the IBM tape subsystem you want to check. Listing the contents of the `/dev/rmt` directory (using the `ls` command) can be helpful in determining the proper special file name. For medium changer devices, the special file name `/dev/rmt/nsmc` should be used.

The following information should be displayed:

```
IBM xxxxxxxx configured at /dev/rmt/nst.
```

where **xxxxxxx** is the model number of the IBM tape subsystem and **n** is the same number specified in the verify command.

To verify that the IBMtape device driver is loaded in kernel memory, enter the following command:

```
/usr/sbin/modinfo | /usr/bin/grep IBMtape
```

The following information should be displayed:

```
165 f5f10000 15c0s 109 1 IBMtape (IBM SCSI Tape/Medium Changer DD)
```

The first five fields shown here usually differ from your specific output. This is not cause for concern. The fields indicate the ID, load address, size, major number, and revision for the IBMtape device driver and vary from system to system.

Problem Determination

If you are experiencing problems with the installation of the IBM SCSI Tape and Medium Changer Device Driver for Solaris, the following information might be of assistance. If you cannot solve your problem after checking the following, contact the appropriate IBM service representative:

Solaris Device Driver (IBMtape)

- If you receive the following message during installation:

```
drvconfig: System call 'modctl_modconfig' failed:
  No such device or address.
Warning: Driver (IBMtape) configuration failed.
  System could not install driver.
```

It indicates that the IBMtape device driver was not loaded because it did not detect the presence of any supported IBM devices on the SCSI bus. Verify that SCSI adapter device driver support is installed and configured correctly. Verify that the IBM tape subsystem is connected to the SCSI bus properly, powered-on, and online. It is not necessary for the tape drive to have a cartridge loaded to be recognized by the IBMtape device driver.

- If you cannot open an IBM device, verify that you are using the correct special file. The IBM tape special files are of the form **st** in the */dev/rmt* directory. The IBM medium changer special files are of the form **smc* in the */dev/rmt* directory. Ensure that the Sun native tape device driver (*st*) is not contending for the same IBM device by consulting the *st.conf* file in the */kernel/drv* directory and commenting out conflicting stanzas.

Downloading Device Microcode

The IBM 3580 Ultrium Tape Drive supports downloading a new microcode image from the host system through the SCSI bus.

To download microcode to this device, perform the following steps:

1. Verify that the IBM tape subsystem is powered-on and online.
2. Verify that the tape drive does not have a tape cartridge loaded.
3. Determine the current version of microcode present on the device by entering the following command, substituting **n** with the actual value from the device special file associated with the target device:

```
/opt/IBMtape/tapeutil -f /dev/rmt/nst -o qmc -v
```
4. Determine the source of the new microcode image file (that is, diskette, such as */vol/dev/aliases/floppy0*, or host file, such as */tmp/ucode_image.file*).
5. Enter the following command, substituting **file** with the actual path and filename of the microcode image file and substituting **n** with the actual value from the device special file associated with the destination device:

```
/opt/IBMtape/tapeutil -f /dev/rmt/nst -o dmc -z file -v
```

Note: The microcode download procedure can also be performed using the menu-driven interface of the *tapeutil* program. To call the *tapeutil* program in this format, enter the following command, then select option 1 to open the device, followed by option 4 to query the current device microcode level and option 7 to download new microcode:

```
/opt/IBMtape/tapeutil
```

Forcing and Storing Device Diagnostic Dump

The IBM 3580 Ultrium Tape Drive supports forcing a diagnostic dump and storing that dump to a mounted tape cartridge or to a host system file through the SCSI bus.

To force and store a diagnostic dump on these devices, perform the following steps:

1. Verify that the IBM tape subsystem is powered-on and online.

2. Enter the following command to force the dump, substituting **n** with the actual value from the device special file associated with the target device:

```
/opt/IBMtape/tapeutil -f /dev/rmt/nst -o fdp -v
```

3. Determine the location for the dump to be stored (that is, diskette, such as */vol/dev/aliases/floppy0* or host file, such as */tmp/diag_dump.file*).

4. Enter the following command, substituting **file** with the actual path and filename of the dump destination file, and substituting **n** with the actual value from the device special file associated with the dump device:

```
/opt/IBMtape/tapeutil -f /dev/rmt/nst -o sdp -z file -v
```

5. The dump can also be written to a tape cartridge mounted in the dump device using *tapedrive* for **file**.

Note: The diagnostic dump procedure can also be performed using the menu-driven interface of the *tapeutil* program. To call the *tapeutil* program in this format, enter the following command, then select option 1 to open the device, followed by option 5 to force a dump and option 6 to store the dump:

```
/opt/IBMtape/tapeutil
```

Tape Monitor Daemon (tmd)

The Tape Monitor Daemon is introduced in the version of IBMtape.4.0.9.1 or later. It is designed to run concurrently with the IBMtape driver and to automatically retrieve and store the IBM Ultrium tape drive diagnostic information (drive dump) into the */var/opt/IBMtape* directory. The daemon is automatically started when the driver is installed, even no any tape device is attached on the system. An entry "name="IBMtape" parent="pseudo" instance=16383;" is also entered into the configuration file of */usr/kernel/drv/IBMtape.conf* automatically for the daemon during the IBMtape driver installation.

The following options can be used to configure the tape monitor daemon running it on the command line. Most options can also be specified in the */etc/tmd.conf* configuration file. However, the command line options will override any configuration file options.

- s Stop any currently running instance of the tape monitor daemon.
- r Restart the tape monitor daemon and reload all configuration settings.
- d Turn on drive error diagnostic retrieval and storage.
This option is enabled by default.
- D Turn off drive error diagnostic retrieval and storage.
- p <directory> Specify an alternate directory for the storage of drive diagnostic information. Default directory is */var/opt/IBMtape*
- l <filename> Specify a file for writing daemon related log messages.
By default, the tmd only writes status information to the syslog file of */var/adm/messages*.
- y Turns off writing log messages to syslog.
- z Turn off compression. By default, the tmd will use a form of file compression to reduce the size of stored diagnostic information.

The file name of dump presents some useful information. An example of the dump file is:

```
IBMtape.000001300148.2004-04-09-14:54:14.dump.gz
```

Here, 000001300148 represents the serial number of the tape device, 2004-04-09-14:54:14 is the time stamp for the dump retrieval.

Solaris Device Driver (IBMtape)

A message is also logged in the syslog file of `/var/adm/messages` after a drive dump is retrieved by `tmd`. For example,

```
Apr  9 14:54:21 Java tmd[3279]: Drive dump saved to /var/opt/IBMtape
IBMtape.000001300148.2004-04-09-14:54:14.dump.
```

Tracing Facility

IBMtape incorporates a tracing facility that is useful for performing problem determination. The tracing facility logs diagnostic information to `/var/adm/messages` based on the control variable `IBM_trace`. See “Setting the `IBM_trace` Level” on page 211 for instructions on how to set the trace value.

`IBM_trace` values range from 0–13 and result in posted messages as shown in Table 23. Postings are cumulative; trace level 3 also posts items for levels 2, 1, and 0. A trace value of 2 or 3 is suitable for most normal production environments with little or no degradation of throughput. `IBM_trace` values of 4 and higher increasingly degrade performance and should generally be used only when directed by IBM support personnel.

Table 23. Tracing Facility

Trace Level	Items Traced
0	Hardware sense data and severe error conditions only.
1	Moderate error conditions
2	Device opens and closes. Decoded SCSI command, sense key, ASC and ASCQ for sense data.
3	Additional device open information. SCSI transport packet information for nonzero return codes.
4–13	Increasingly verbose tracing information. These tracing levels are generally useful only to IBMtape developers.

Note: IBMtape Versions before 4.0.2.7 had only `IBM_trace` values 0–4. Message content and selection differed significantly from present IBMtape versions.

By default, system error messages, including IBMtape trace messages, are placed in `/var/adm/messages`. If your installation has modified `/etc/syslog.conf` to redirect system error messages, IBMtape tracing is handled as other kernel messages. See the `syslog.conf` man page and comments in `syslog.conf` for information about the system logging operation. Changes made to `syslog.conf` take effect after the next system reboot.

Following is a sample of trace level 2 output with system date and time stamps removed. Device instance 36 is opened on the first line. The device minor number 0x4C4 is decoded and shows that the SCSI tape drive (`drv`) special file was opened.

The second line decodes selected fields from the sense data, which immediately follows. Looking up the decoded Sense Key /ASC/ASCQ combination in the 3580 hardware reference, we find that the command failed because a file mark was encountered during the space.

The actual sense data follows the decoded fields.

Note: Solaris, rather than printing multiple 16-byte lines of hex zeros, instead prints only the first such line, followed by a repeat count.

```

IBMtape( 36) _open: Ins36 Mnr0x4c4<BSD,NoRew>(drv)
                    Flg0x5<Ndelay,Read> TL24036
IBMtape( 36) check_sense: cmd 0x11(space) , key/asc/ascq 0x0/0/1,
                    defer 0, retry 0, rc 5
IBMtape( 36) ULTRIUM-TD1      S/N 1300015708 SENSE DATA
IBMtape( 36) f0 0 80 0 0 0 1 1c 0 0 0 0 0 1 0 0
IBMtape( 36) 0 0 0 0
IBMtape( 36) _close: Inst 36, Minor 1220 (drv), Flags 0x5, exit(0)

```

The following sense data for device instance 36, a tape drive, occurred during a test unit ready and indicates that a tape is in the throat of the drive but requires an Initializing Command (that is, a SCSI Load command) to move the tape fully into the drive.

```

IBMtape( 36) _open: Ins36 Mnr0x4c4<BSD,NoRew>(drv)
                    Flg0x5<Ndelay,Read> TL24036
IBMtape( 36) check_sense: cmd 0x0(test_unit_ready) , key/asc/ascq 0x2/4/2,
                    defer 0, retry 0, rc 5
IBMtape( 36) ULTRIUM-TD1      S/N 1300015708 SENSE DATA
IBMtape( 36) 70 0 2 0 0 0 0 1c 0 0 0 0 4 2 0 0
IBMtape( 36) 10 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0
IBMtape( 36) 0 0 0 0
IBMtape( 36) 10 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0
IBMtape( 36) 0 0 0 0
IBMtape( 36) _close: Inst 36, Minor 1220 (drv), Flags 0x5, exit(0)

```

You can match an instance number with its corresponding device special files in two steps:

1. Find the instance number in `/etc/path_to_inst`:

```

$ grep 292 /etc/path_to_inst
"/pci@6,4000/scsi@2,1/IBMtape@2,0" 292 "IBMtape"

```

2. List “long” the contents of `/dev/rmt` and search for the path name you found in the previous step:

```

$ ls -l /dev/rmt | grep "/pci@6,4000/scsi@2,1/IBMtape@2,0"
lrwxrwxrwx 1 root other 48 Aug 26 11:49 8st ->
  ../../devices/pci@6,4000/scsi@2,1/IBMtape@2,0:st
lrwxrwxrwx 1 root other 49 Aug 26 11:49 8stb ->
  ../../devices/pci@6,4000/scsi@2,1/IBMtape@2,0:stb
.
.
.

```

In this example, `/dev/rmt/8st`, `/dev/rmt/8stb`, and so on, are symbolic links to the device special files that are associated with device instance 292.

Setting the IBM_trace Level

The default value for `IBM_trace` is zero (0). You can define another `IBM_trace` value by placing an entry in `/etc/system`, so that `IBM_trace` is set at each reboot. For example, this entry in `/etc/system` sets `IBM_trace` to 2 at each reboot:

```
set IBMtape:IBM_trace = 2
```

When `IBM_trace` is set in `/etc/system`, it affects tracing during driver loading, initialization, and operation.

You can also set or modify the `IBM_trace` value manually in an `adb` session. Because the driver must already be loaded and initialized before using this method, the trace value that is set is active only during driver operation.

Solaris Device Driver (IBMtape)

In this sample session, *ksh*> is a shell prompt, and *adb*> is the *adb* session prompt. Commands that you enter follow these prompts. Explanatory comments follow pound signs (#) or exclamation and pound sign pairs (!#). Text lines without a prefix are *adb* session responses to commands.

```
#
# Start adb session and set session prompt.
ksh> adb -P "adb> " -k -w /dev/ksyms /dev/mem
physmem 7c5e
!#
!# Set default for input values to base 10.
adb> a$d
radix=10 base ten
!#
!# Display current IBM_tape value as unsigned decimal integer.
adb> IBM_trace/u
IBM_trace:
IBM_trace:      0
!#
!# Set new IBM_trace value.
!# adb will confirm the old and new values.
adb> IBM_trace/w 2
IBM_trace:      0      =      2
!#
!# Quit session.
adb> $q
#
# Back to the shell.
ksh>
```

Running the Diags_info Script

Run the *diags_info* script located in the */opt/IBMtape* directory. This script detects the problems on the configuration files, gather important system HBAs and configuration information. The script should be run as root. If it is not run as root, the information should be labeled as such, but the value of the information is degraded when run as a non-root user.

To facilitate capture of data, the script places information in a file called *diags.out* in the directory locating the script. Send the output file to the location identified by your IBM service representative.

Tape and Medium Changer Utility Program

A SCSI Tape and Medium Changer Utility Program called *tapeutil* is provided with the IBM SCSI Tape Drive and Medium Changer Device Driver for Solaris and is installed in the */opt/IBMtape* directory as part of the IBMtape package. This program fulfills several purposes:

- It provides the following service aids for IBM tape subsystems:
 - Query Device Type/Verify Device Attachment
 - Query Device Serial Number
 - Query Device Microcode Level
 - Force Device Diagnostic Dump
 - Store Device Diagnostic Dump
 - Download Device Microcode
- It provides a menu-driven test tool for exercising or testing IBM tape and medium changer devices with a full suite of supported operations:
 - Reading/Writing Data

- Tape Motion Commands
- Setting/Displaying Device Information/Status
- Mounting/Demounting Cartridges
- Cartridge Inventory
- In addition to the menu-driven front end, the *tapeutil* program provides a command-line interface that is convenient for use in shell scripts.

Note: When using the command-line calls to the *tapeutil* program, the tape device is opened and closed for each invocation. Configuration parameters that are changed in one call to the *tapeutil* program are returned to default values when the device is closed.

- The source code for the *tapeutil* program is provided for example purposes and is installed in the */opt/IBMtape* directory during the IBMtape package installation. This source code is commented and demonstrates calls to all of the supported device driver entry points and *ioctl* commands, thus giving the application developer a starting point for interfacing to the IBMtape device driver.

The *tapeutil* program provides both an interactive menu-driven interface and a command-line interface. If the *tapeutil* program is called with no command-line parameters, the menu-driven version is started. In the menu-driven version, the device to be operated on should first be opened using option 1. Other options can then be selected. The user is prompted for additional information if required for the specific options selected. The results of a command are displayed after it is executed. If an error occurs for the command, error information and device sense data are displayed. The device can be closed using option 2, or it is closed automatically when the Quit option is selected. The menu is displayed once automatically when the program is first called. To prevent unnecessary scrolling of the screen, the menu is not displayed automatically again after each command but is instead refreshed only after the M (menu refresh) command is entered.

Figure 10 and Figure 11 on page 214 show the menus that are displayed by the *tapeutil* program.

IBM SCSI TAPE & MEDIUM CHANGER UTILITY PROGRAM		
<< GENERAL COMMANDS >>	<< SERVICE COMMANDS >>	<< BASIC SCSI COMMANDS >>
1: Open Device	3: Serial/Subsys Number	10: Inquiry
2: Close Device	4: Query Microcode Level	11: Request Sense
52: Get Driver Info	5: Force Dump	12: Reserve
D: Device Type	6: Store Dump	13: Release
M: Menu Refresh	7: Download Microcode	50: Log Sense Page
Q: Quit Program	9: Test Unit Ready	51: Mode Sense Page
<< TAPE DRIVE COMMANDS >>		
20: Read Data	29: Backward Space File	38: Get Position
21: Write Data	30: Forward Space Rec	39: Set Position
22: Write File Mark	31: Backward Space Rec	42: Sync Buffer
23: Erase Tape	32: Locate End of Data	43: Display Message
24: Rewind	33: Get Record Size	45: Report Density
25: Retension	34: Set Record Size	48: Query/Set Parameters
26: Offline	35: Get Device Status	49: Read/Write Tests
27: Load/Unload Tape	36: Get Device Info	
28: Forward Space File	37: Get Media Info	

Figure 10. TAPEUTIL Program Menu for the Tape Drive

Solaris Device Driver (IBMtape)

IBM SCSI TAPE & MEDIUM CHANGER UTILITY PROGRAM		
<< GENERAL COMMANDS >>	<< SERVICE COMMANDS >>	<< BASIC SCSI COMMANDS >>
1: Open Device	3: Serial/Subsys Number	10: Inquiry
2: Close Device	4: Query Microcode Level	11: Request Sense
52: Get Driver Info	5: Force Dump	12: Reserve
D: Device Type	6: Store Dump	13: Release
M: Menu Refresh	7: Download Microcode	50: Log Sense Page
Q: Quit Program	9: Test Unit Ready	51: Mode Sense Page
<< MEDIUM CHANGER COMMANDS >>		
14: Move Medium	17: Inventory	46: Read Device IDs
15: Position To Element	18: Audit	47: Audit Range
16: Element Information	19: Lock/Unlock Door	

Figure 11. TAPEUTIL Program Menu for the Medium Changer

If command-line parameters are provided when the program is called, the command-line mode is started. For each command-line execution of the program, the device is first opened, the specific command is issued, then the device is closed. The program can be driven from within a shell script if desired. Results of the operation are displayed only when executed in verbose mode. No information is displayed when not in verbose mode. This is particularly useful for quiet shell script implementations. A completion code, as defined in `/usr/include/sys/errno.h`, for the operation requested is always returned from the program upon exit (in both verbose and quiet mode).

The usage of the `tapeutil` program in command-line mode is as follows:

```
tapeutil -f device -o operation [options]
```

Where **device** is the name of the tape device special file (for example: `/dev/rmt/1st`) and **operation** is one of the following values. The device special file and the operation are required. The specific **options** associated with a particular operation are indicated. Parameters enclosed in square brackets are optional. All others are required.

Service Commands

Query Serial Number	<code>tapeutil -f f -o qsn [-w w] [-v]</code>
Query Microcode Level	<code>tapeutil -f f -o qmc [-w w] [-v]</code>
Force Dump	<code>tapeutil -f f -o fdp [-w w] [-v]</code>
Store Dump	<code>tapeutil -f f -o sdp [-w w] [-v] -z z</code>
Download Microcode	<code>tapeutil -f f -o dmc [-w w] [-v] -z z</code>
Format Cartridge	<code>tapeutil -f f -o fmt [-w w] [-v]</code>
Query Device Type	<code>tapeutil -f f -o chk [-w w] [-v]</code>

Basic SCSI Commands

Test Unit Ready	<code>tapeutil -f f -o tur [-w w] [-v]</code>
Inquiry	<code>tapeutil -f f -o inq [-w w] [-v] -t t -x x</code>
Request Sense	<code>tapeutil -f f -o req [-w w] [-v]</code>

Reserve	<code>tapeutil -f f -o res [-w w] [-v]</code>
Release	<code>tapeutil -f f -o rel [-w w] [-v]</code>

Medium Changer Commands

Move Medium	<code>tapeutil -f f -o mov [-w w] [-v] -s s -d d</code>
Position To Element	<code>tapeutil -f f -o pos [-w w] [-v] -d d</code>
Element Information	<code>tapeutil -f f -o ele [-w w] [-v]</code>
Inventory	<code>tapeutil -f f -o inv [-w w] [-v]</code>
Audit	<code>tapeutil -f f -o aud [-w w] [-v]</code>
Lock/Unlock Door	<code>tapeutil -f f -o lck [-w w] [-v] -x x</code>

Tape Drive Commands

Read	<code>tapeutil -f f -o rea [-w w] [-v] -b b -n n -m m</code>
Write	<code>tapeutil -f f -o wri [-w w] [-v] -b b -n n -m m</code> <code>[-r r] [-z z]</code>
Write File Mark	<code>tapeutil -f f -o eof [-w w] [-v] -c c</code>
Erase Tape	<code>tapeutil -f f -o era [-w w] [-v]</code>
Rewind	<code>tapeutil -f f -o rew [-w w] [-v]</code>
Retension	<code>tapeutil -f f -o ret [-w w] [-v]</code>
Offline	<code>tapeutil -f f -o off [-w w] [-v]</code>
Load/Unload Tape	<code>tapeutil -f f -o lod [-w w] [-v] -x x</code>
Forward Space File	<code>tapeutil -f f -o fsf [-w w] [-v] -c c</code>
Backward Space File	<code>tapeutil -f f -o bsf [-w w] [-v] -c c</code>
Forward Space Record	<code>tapeutil -f f -o fsr [-w w] [-v] -c c</code>
Backward Space Record	<code>tapeutil -f f -o bsr [-w w] [-v] -c c</code>
Locate End of Data	<code>tapeutil -f f -o eod [-w w] [-v]</code>
Get Record Size	<code>tapeutil -f f -o grs [-w w] [-v]</code>
Set Record Size	<code>tapeutil -f f -o srs [-w w] [-v] -x x</code>
Get Device Status	<code>tapeutil -f f -o gds [-w w] [-v]</code>
Get Device Information	<code>tapeutil -f f -o gdi [-w w] [-v]</code>
Get Media Information	<code>tapeutil -f f -o gmi [-w w] [-v]</code>
Get Position	<code>tapeutil -f f -o gpo [-w w] [-v] -t t</code>
Set Position	<code>tapeutil -f f -o spo [-w w] [-v] -t t -x x</code>
Get Parameter	<code>tapeutil -f f -o gpa [-w w] [[-v] -t t</code>
Set Parameter	<code>tapeutil -f f -o spa [[-w w] [-v] -t t -x x</code>
Sync Buffer	<code>tapeutil -f f -o syn [-w w] [-v]</code>
Display Message	<code>tapeutil -f f -o msg [-w w] [-v] -t t -y y1,y1</code>

Note: Calling the *tapeutil* program with the **-h** flag (for example, **tapeutil -h**) or the **-?** flag (for example, **tapeutil -?**) displays the usage help information.

Flag Description

The following are the supported flags, their meanings, their associated operations, and their acceptable ranges:

Flag	Description
-?	Usage Help (stand-alone flag) {no value required}
-b	Block Size (rea, wri) {0 < (block size x blocking factor) < 2097152}
-c	Operation Count (eof, fsf, fsr, bsf, bsr) {0–65535}
-d	Destination Address (mov) {device specific, determine range from Element Info}
-f	Device Special File Name (always required) {/dev/rmt/1st or similar}
-h	Usage Help (stand-alone flag) {no value required}
-m	Multiples to Read or Write (rea, wri) {0–2097152}
-n	Blocking Factor (rea, wri) {0 > (block size x blocking factor) < 2097152}
-o	Operation (always required) {see previous list}
-r	Random Seed (wri) {0–65535}
-s	Source Address (mov, pos) {device specific, determine range from Element Info}
-t	Type of Parameter Value <ul style="list-style-type: none">• (gpo) {1=logical block, 2=physical block}• (spo) {1=logical block, 2=physical block}• (gpa) {1=block size, 2=compression, 3=buffering, 4=immediate, 5=trailer, 6=write protect, 7=acf mode, 8=capacity, 9=sili}• (spa) {1=block size, 2=compression, 3=buffering, 4=immediate, 5=trailer, 6=write protect, 8=capacity, 9=sili}• (msg) {1=display msg0, 2=display msg1, 3=flash msg0, 4=flash msg1, 5=alternate msg1/msg2}• (inq) {0=standard data, 1=page data}
-v	Verbose Mode (optional for all commands, stand-alone flag) {no value required, absence of flag means quiet mode}
-w	Open Mode (optional for all commands) {1=read/write, 2=read only (default), 3=write only, 4=append}
-x	Parameter Value

- (lck) {1=lock, 2=unlock}
 - (lod) {1=load, 2=unload}
 - (inq) {0x0-0xFF}
 - (srs) {0-65536}
 - (spo) {0-65535}
 - (spa) {0-65535}
- y Messages (msg) {message1,message2}
- z Input/Output File Name
- (sdp) {path and name of the file in which to store dump}
 - (dmc) {path and name of the microcode image file}
 - (wri) {path and name of the file containing write data pattern}

Notes:

1. For read and write operations, the size of one buffer of data transferred during a single SCSI read or write command is determined by the product of the *Block Size* value and the *Blocking Factor* value. The number of these buffers transferred is determined by *Multiplier* value. The actual total number of bytes transferred is then (Block Size) x (Blocking Factor) x (Multiplier). If the device is set to fixed block mode (block size not equal to zero), the product of *Block Size* and *Blocking Factor* must be a multiple of the device block size setting.
2. For further information about the Get Parameter (gpa) and Set Parameter (spa) operations, see the STIOC_GET_PARM and STIOC_SET_PARM *ioctl* commands described in the *IBM Ultrium Device Drivers: Programming Reference*.
3. For 3592 media types, capacity scaling is only supported for the standard 300 GB rewriteable data cartridge.
4. Any attempt to set capacity scaling that is not supported by a device or the current media loaded will always return 100% and cannot be changed. For example, the 60GB (Economy Data) cartridge for the IBM 3592 cannot be capacity scaled and will always be 100%.

The following examples should help to demonstrate and clarify the command-line usage of the *tapeutil* program. For all examples, substitute the actual value of the special file associated with the target device.

- To query the serial number of the device:
/opt/IBMtape/tapeutil -f /dev/rmt/0st -o qsn -v
- To request inquiry data from the device:
/opt/IBMtape/tapeutil -f /dev/rmt/0st -o inq -v
- To request inquiry page data from the device:
/opt/IBMtape/tapeutil -f /dev/rmt/0st -o inq -t 1 -x 0x83 -v
- To move a cartridge from cell 32 to the tape drive (16):
/opt/IBMtape/tapeutil -f /dev/rmt/0smc -o mov -s 32 -d 16 -v
- To set the block size of the device to 64K:
/opt/IBMtape/tapeutil -f /dev/rmt/0st -o spa -t 1 -x 65535 -v
- To write 100 64K blocks of data to the tape device:
/opt/IBMtape/tapeutil -f /dev/rmt/0stn -w 1 -o wri -b 65535
-n 1 -m 100 -v
- To write two file marks to the tape device:
/opt/IBMtape/tapeutil -f /dev/rmt/0stn -w 1 -o eof -c 2 -v
- To rewind the tape device:

Solaris Device Driver (IBMtape)

```
/opt/IBMtape/tapeutil -f /dev/rmt/0stn -o rew -v
```

- To read 100 64K blocks of data from the tape device:

```
/opt/IBMtape/tapeutil -f /dev/rmt/0stn -o rea -b 65535 -n 1 -m 100 -v
```

Tapelist Utility Program (tapelist)

A Tapelist Utility Program called *tapelist* is installed in the `/opt/IBMtape` directory as part of the IBMtape package, after you install the version of IBMtape.4.0.9.1 or later. The tapelist utility provides the user a listing of tape, medium changer and SAN data gateway devices currently configured by the IBMtape driver.

```
# tapelist
Instance : 697
Special File : /dev/rmt/6st
Device : 03592J1A
Serial Number : 000001300168
TGT/LUN : 7/0
Ucode : 04C4
World Wide NN : 5005076302000127
World Wide PN : 5005076302400127
Dev Phy Path : /devices/pci@1f,2000/QLGC,qla@1/IBMtape@7,0:st
Path Type : N/A
```

The following headers appear over various columns of data in this example list:

Inst #	The instance number of the particular device.
Special File	The device special file used to access this device.
Device	A string indicating the device model.
Serial No	The serial number of the device.
TGT/LUN	The SCSI target and LUN of the device.
Ucode level	The current microcode (firmware) loaded on the device.
World Wide NN	A number indicating Fibre Channel World Wide Node Name of the device.
World Wide PN	A number indicating Fibre Channel World Wide Port Name of the device.
Dev Phy Path	A string indicating the device path in the device tree.
Path Type	A primary or alternate path used for failover.

The usage of the tapelist program is as follows:

```
-l Print for all of the configured devices with the column headers in long list
-c Don't print column headers in long list for all of the configured devices
-f Print the list for a particular file only
-h Help menu
```

Running tapelist without any options displays the device information line by line for all of the configured devices.

Part 7. Microsoft Windows Tape Device Drivers

Chapter 40. Introduction and Product Requirements

Windows NT

Hardware Requirements

The Windows NT[®] device driver supports the following IBM Ultrium tape drives and automation products:

- One or more of the following IBM Ultrium tape devices:
 - IBM TotalStorage Ultrium External Tape Drive 3580
 - IBM TotalStorage Ultrium Tape Autoloader 3581
 - IBM TotalStorage Ultrium Tape Library 3582
 - IBM TotalStorage Ultrium Scalable Tape Library 3583
 - IBM TotalStorage UltraScalable Tape Library 3584
- One or more of the following SCSI host adapters:
 - Adaptec 2940U2W or IBM P/N 33L5000 for IBM xSeries (LVD)
 - Adaptec 2944UW (HVD)
 - Adaptec SCSI Adapter 29160 or IBM P/N 19K4646 for IBM xSeries - Single Port LVD
 - Adaptec SCSI Adapter 39160 - Dual Port LVD
 - Adaptec SCSI Card 39320-R (LVD) and 39320D-R (dual port LVD)
 - Symbios SYM22910 64-bit PCI-to-Ultra-2 SCSI Dual Channel Host Adapter (LVD) from LSI Logic Corporation
- One or more of the following FC-AL host bus adapters:
 - QLogic QLA2200F, QLA2310FL, QLA2340, QLA2340L, QLA2342, QLA2342L Fibre Channel Adapters
 - Emulex LightPulse LP8000 and LP9002 Fibre Channel Adapters
 - TotalStorage FastT FC-2 and FC2-133 FC Host Bus Adapters
 - Emulex LP9002L
 - Emulex LP952L
 - Emulex LP9402DC
 - Emulex LP9802
 - Emulex LP10000
- The IBM 2108-G07 (IBM SAN Data Gateway) and IBM 2108-R03 (IBM SAN Data Gateway Router) can be used to attach SCSI models of the IBM Ultrium family of products with any of the supported FC host bus adapters

The IBM Ultrium tape drives and automation products are supported on Intel-compatible processors with a minimum processor level of Intel 486DX or Pentium[®] with sufficient RAM and disk space for operation of the Microsoft Windows NT operating system.

Windows Device Drivers (IBMtape)

Attention: Using a single Fibre Channel host bus adapter (HBA) for concurrent tape and disk operations is not recommended. Tape and disk devices require incompatible HBA settings for reliable operation and optimal performance characteristics. Under stress conditions (high I/O rates for tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems have been observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and using SAN zoning to minimize contention. IBM is focused on assuring server and storage configuration interoperability. It strongly recommends that your implementation plan includes provisions for separating disk and tape workloads.

Software Requirements

The software requirements are:

- SCSI or FC adapter device driver (typically shipped with the adapter or resident in Windows NT)
- Microsoft Windows NT Version 4.0 with Service Pack 6 or later installed

To obtain the most current service and documentation for this software, see Appendix A, “Accessing Documentation and Software Online,” on page 287.

Installation Notes[®]

- To determine the Windows NT Version and Service Pack level, open the Control Panel, click **Help**, then click **About Windows NT**.
- The system bus is scanned for devices only at boot time. In order to use your Ultrium devices, they must be connected, powered-on, and enabled when the system is booted.
- To verify your connection during boot, you should see the IBM Ultrium device being detected by the SCSI or FC adapter. A message such as one of the following should be displayed:

```
Adaptec SCSI Card 39160 Bios V2.57.0(c) 2000 Adaptec, Inc. All Rights Reserved.
Ch A, SCSI ID:0      IBM Ultrium:Td1      80.0
```

or

```
Symbios, Inc SDMS™ V4.0 pci scsi bios, pci rev. 2.0.2.1
```

```
Copyright 1995, 1998 Symbios, Inc
```

```
PCI-4.14.00
```

HBA	LD	LUN	Vendor	PRODUCT	Rev	SYNC	Wide
0	0	0	IBM	Ultrium-TDI	0610	Yes	16

Note: The IBM 3584 UltraScalable Tape Library will not be displayed on the Adaptec SCSI cards list of detected devices during the boot.

- After the device driver is installed:

If your devices are not connected, powered On, and Online when the system is booted, a **Service Control Manager** window will open. You will receive this message:

At least one service or driver failed during system startup. Use Event Viewer to examine the event log for details.

- When the Windows NT device driver and media mover device drivers start, they manage all IBM Ultrium devices and changers exclusively. Other device drivers on the system that access IBM Ultrium devices could cause conflicts and indeterminate behavior. Before installing and starting the Windows NT device driver and media mover device drivers, ensure that all other device drivers are uninstalled or disabled. Rebooting the system after removal of the other device drivers is recommended.

A customer in an environment where more than one device driver is required with Ultrium devices and changers might want to start and stop these device drivers manually at the appropriate times rather than install and remove the device drivers. See “Manual Starting and Stopping Procedures” on page 236 for details.

Windows 2000 and Windows Server 2003

The Windows 2000 and Windows Server 2003 drivers are very similar. Common documentation for these similar devices will be indicated by Windows 200x.

Hardware Requirements

The Windows 200x device driver supports the following IBM Ultrium tape drives and automation products:

- One or more of the following IBM Ultrium tape devices:
 - IBM TotalStorage Ultrium External Tape Drive 3580
 - IBM TotalStorage Ultrium Tape Autoloader 3581
 - IBM TotalStorage Ultrium Tape 2U Autoloader 3581
 - IBM TotalStorage Ultrium Tape Library 3582
 - IBM TotalStorage Ultrium Scalable Tape Library 3583
 - IBM TotalStorage UltraScalable Tape Library 3584
- One or more of the following SCSI host adapters:
 - Adaptec 2940U2W or IBM P/N 33L5000 for IBM xSeries (LVD)
 - Adaptec 2944UW (HVD)
 - Adaptec SCSI Adapter 29160 or IBM P/N 19K4646 for IBM xSeries - Single Port LVD
 - Adaptec SCSI Adapter 39160 - Dual Port LVD
 - Adaptec SCSI Card 39320-R (LVD) and 39320D-R (dual port LVD)
 - Symbios SYM22910 64-bit PCI-to-Ultra-2 SCSI Dual Channel Host Adapter (LVD) from LSI Logic Corporation
- One or more of the following FC-AL host bus adapters:
 - QLogic QLA2200F, QLA2310FL, QLA2340, QLA2340L, QLA2342, QLA2342L Fibre Channel Adapters
 - Emulex LightPulse LP8000 and LP9002 Fibre Channel Adapters
 - TotalStorage FastT FC-2 and FC2-133 FC Host Bus Adapters
- With the Microsoft Removable Storage Manager disabled, the medium changers of the Ultrium family of products are supported through the IBM 2108-G07 (IBM SAN Data Gateway) and IBM 2108-R03 (IBM SAN Data Gateway Router) with any of the supported FC host bus adapters, except the Emulex LP9002L. The customer will need to provide an application to operate the medium changer.

The drives of the IBM Ultrium family of products are supported through the IBM 2108-G07 (IBM SAN Data Gateway) and IBM 2108-R03 (IBM SAN Data Gateway Router) with any of the supported FC host bus adapters

The IBM Ultrium tape drives and automation products are supported on Intel-compatible processors with sufficient RAM and disk space to run Microsoft Windows 2000 Build 2195, Windows Server 2003 Build 3790, or later.

Attention: Using a single Fibre Channel host bus adapter (HBA) for concurrent tape and disk operations is not recommended. Tape and disk devices require incompatible HBA settings for reliable operation and optimal performance characteristics. Under stress conditions (high I/O rates for tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems have been observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and using SAN zoning to minimize contention. IBM is focused on assuring server and storage configuration interoperability. It strongly recommends that your implementation plan includes provisions for separating disk and tape workloads.

Software Requirements

The software requirements are:

- SCSI or FC adapter device driver (typically shipped with the adapter or resident in Windows 200x)
- Microsoft Windows 2000 Build 2195, Windows Server 2003 Build 3790, or later installed

To obtain the most current service and documentation for this software, see Appendix A, “Accessing Documentation and Software Online,” on page 287.

Installation Notes

These procedures assume that a supported host bus adapter has been installed and configured already.

The recommended procedure is to install the device drivers before installing any Ultrium devices on the SCSI bus.

There might be a noticeable delay before the Windows 200x plug-and-play manager recognizes new devices.

For information on the Removable Storage Manager and Windows 200x Media Services:

1. Click **My Computer**.
2. Click **Control Panel**.
3. Click **Administrative Tools**.
4. Open the **Computer Management** window.
5. Click **Storage**.
6. Click **Actions**.
7. Click **Help** in the pulldown menu.
8. Click **Removable Storage** in the **Help** window that opens.

These drivers conform to the Microsoft SDK Tape and NTMS APIs, as described in Microsoft Platform SDK Windows 200x documentation.

Tape functions are described in the SDK documentation:

- Base Services
- Files and I/O
- Tape Backup

NTMS Services are described in the SDK documentation:

Windows Device Drivers (IBMtape)

- Base Services
- Removable Storage Manager

Changer functions are described in the DDK documentation:

- Kernel Mode Drivers (Reference Part 1, Section 5)

Utilization of these drivers requires specific knowledge of the operation of Ultrium devices as described in the appropriate Hardware Reference Manuals.

This device driver allows the Windows 200x Removable Storage Manager to manage Ultrium devices. It could cause conflicts and indeterminate behavior if used with other device drivers that manage Ultrium devices or with products that use other device drivers for Ultrium.

Chapter 41. Windows NT Device Driver Management

This chapter describes how to install, remove, start, and stop the Windows NT Tape Device Driver for the Ultrium devices.

Installation Overview

The installation process consists of the following steps:

1. Verify that the prerequisites have been satisfied.
2. Install the SCSI or FC adapter.
3. Install the SCSI or FC adapter device driver.
4. Connect the Ultrium device to the adapter.
5. Power on the Ultrium device.
6. Reboot the system.
7. Create an emergency repair disk (optional) using the following procedure:
 - a. Select **Start, Help, and Find** from the Windows NT desktop.
 - b. Type **Emergency** in box 1.
 - c. Select **Using the Repair Disk utility to make an emergency repair disk**.
 - d. Click **Display**, then follow the instructions to make an emergency repair disk (ERD).

Note: See “Software Requirements” on page 223.

This step is highly recommended.

8. Install the Ultrium Tape device driver.

Installation Procedure

To install the device drivers, follow this procedure:

1. Log on as Administrator.
2. Insert the Tape Device Drivers and SMI-S Agent CD in the CD-ROM drive.
3. Go to the **Drivers/Windows/WinNT** directory and double-click **IBMUltrium.WinNT.exe**.
4. Follow the *InstallShield* direction to install the package.
 - a. If you select **compact** installation, the program will copy the system files of the latest IBM Ultrium Device Driver (**Device Driver System Files** component) to your system directory. This provides all the latest support necessary for Ultrium devices.

Note: This option overwrites previous versions of Ultrium device driver installed on your system.

- b. If you select **typical** installation, all support included in the **compact** installation will be done. Also, the **Device Driver Depot** component will be installed, which includes copying all available versions included in the *InstallShield* package to a user-defined directory.

Note: This option overwrites previous versions of Ultrium device driver installed on your system.

Windows NT Device Driver (IBMtape)

- c. If you select **custom** installation, the program will allow you to select the components to install (see Figure 12).

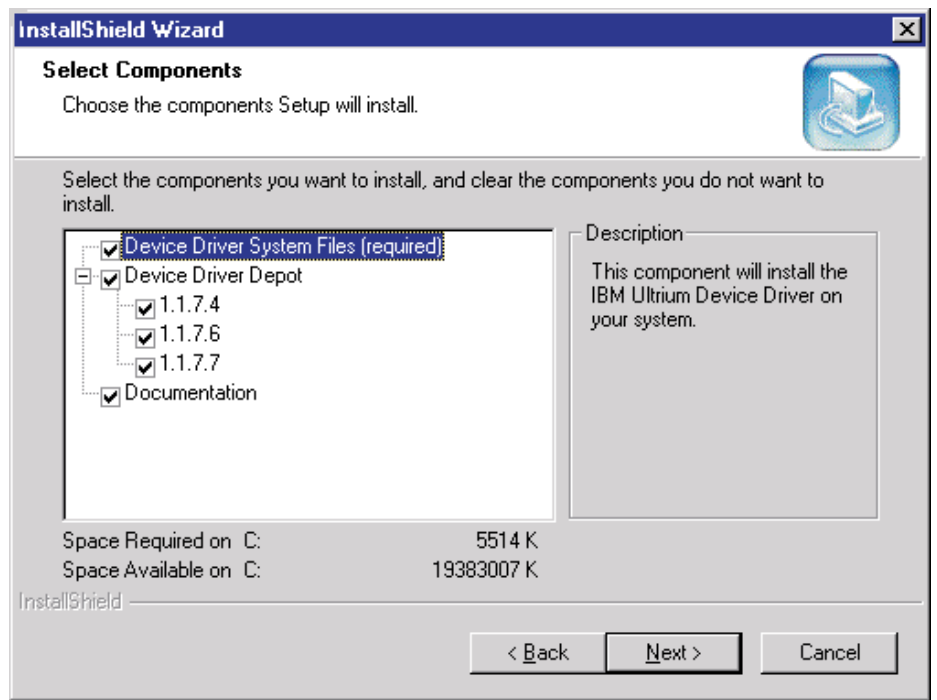


Figure 12. Select Components Menu

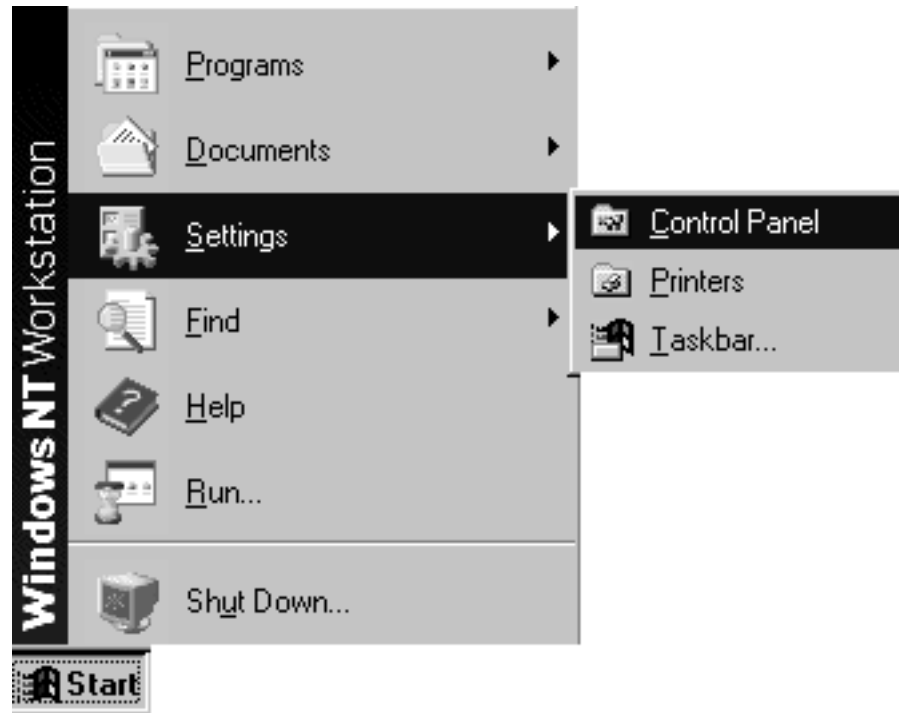
The **Device Driver Depot** component includes the most recent versions of the device drivers. You can select which versions of the device driver to install. The selected versions will not be installed to the system directories and will be dormant. If you wish to activate a version of the driver other than the most recent version, you will need to perform the following steps after the installation is complete:

- 1) Using Windows Explorer, navigate to *c:\Program Files\IBM Corporation\IBM Ultrium Device Drivers* and select the folder for the version you wish to install.
- 2) Copy all files with the *.sys* extension to *c:\winnt\system32\drivers*.
- 3) Copy all files with the *.inf* extension to *c:\winnt\inf*.
- 4) Copy *ntutil.exe* to *c:\winnt\system32*.
- 5) Proceed with these instructions to configure the drivers.

The **Documentation** component copies the PDF version of the *IBM Ultrium Device Drivers: Installation and User's Guide* and the *IBM Ultrium Device Drivers: Programming Reference* to your hard drive.

The **Device Driver System Files** component copies all the files needed for device driver support to the system directory.

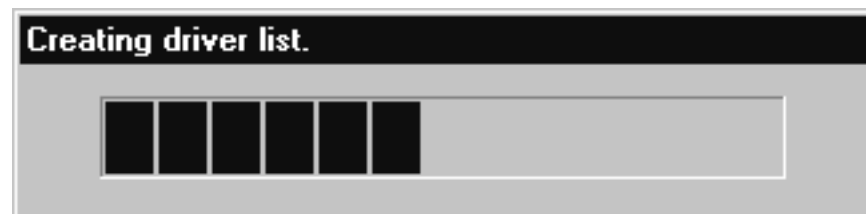
5. After you have completed installing *IBM Ultrium Device Driver for Microsoft Windows NT 4.0*, proceed to enable the driver. Click **Start**, move to **Settings**, then click **Control Panel**. See Figure 13 on page 229.



A250118

Figure 13. Start Menu

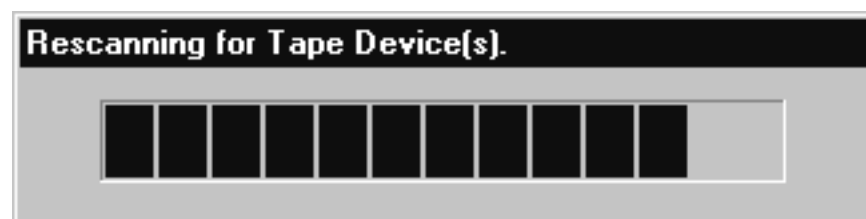
6. Double-click **Tape Devices**. If the Ultrium tape or changer device was already powered On and attached to the system during boot up, the devices should be in the box, and Windows NT should start to create the driver list. See Figure 14.



A250119

Figure 14. Start Driver List

If Windows NT did not detect the attached Ultrium device, click **Detect** to select the device, and Windows NT will rescan the bus. See Figure 15.



A250120

Figure 15. Rescan for Tape Device

Windows NT Device Driver (IBMtape)

If you still cannot see the device, ensure that the cable is attached properly. Also, ensure that the device is terminated properly, the device is powered On, and the adapter driver is enabled.

If you have more than one IBM Ultrium device attached on the same host system, the operating system will prompt you multiple times for the same driver. **Cancel** the multiple driver installation requests.

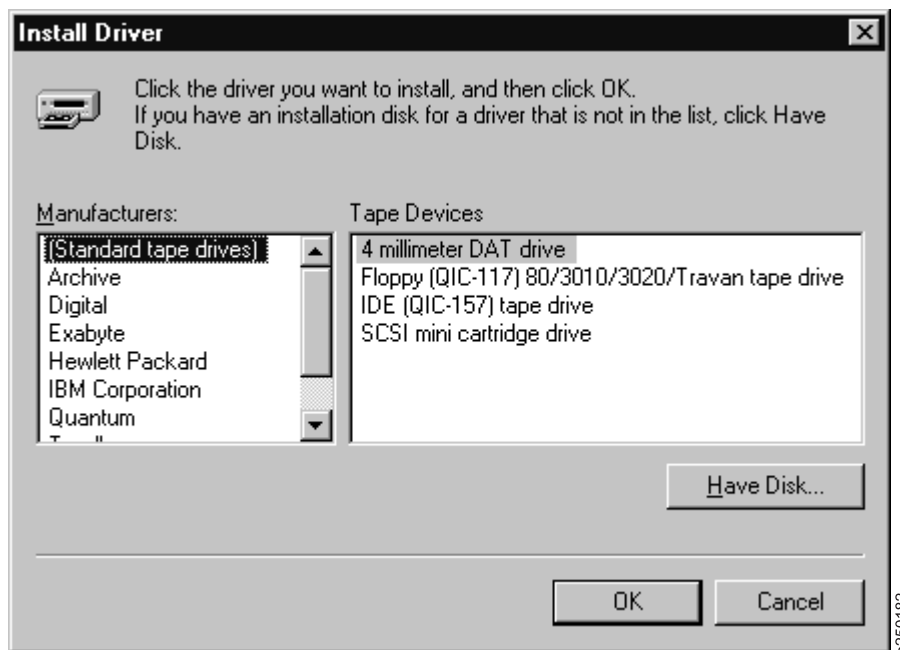


Figure 16. Install Driver Menu — Select Cancel

Click the **Drivers** tab, then click **Add...** See Figure 17 on page 231.

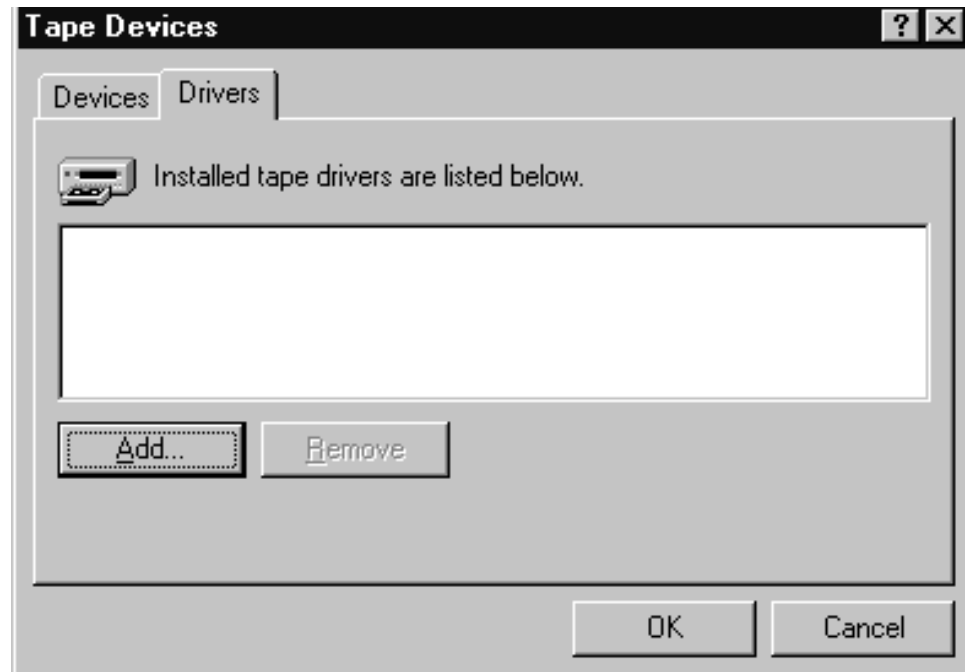
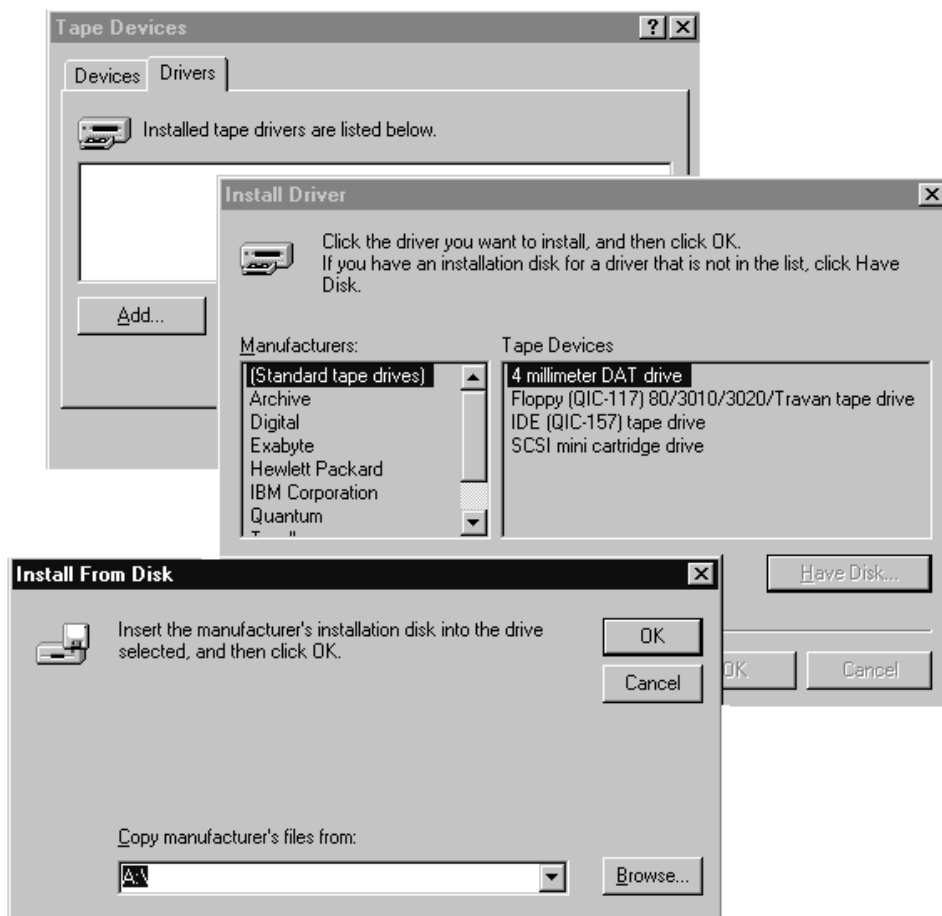


Figure 17. Tape Devices Menu

7. If you installed the **Device Driver System Files** component in step 4 on page 227, **skip to step 8** on page 232. If you did not install the **Device Driver System Files** component, click **H**ave Disk... and enter the directory where your device driver setup file (*IBMUltrium.inf*) is located in the **C**opy manufacturer's files from: box. You can also click **B**rowse... to select the directory. See Figure 18 on page 232.

Windows NT Device Driver (IBMtape)



A250122

Figure 18. Have Disk Menu

8. Click **IBM Corporation**, select the device driver that matches your tape device, then click **OK**. See Figure 19 on page 233.

Note: If you are using both a tape drive and a medium changer (as you would find in a tape library, for example), select one of the **IBM Ultrium Tape Libraries**. This will install drivers for both the medium changer and the tape drives.

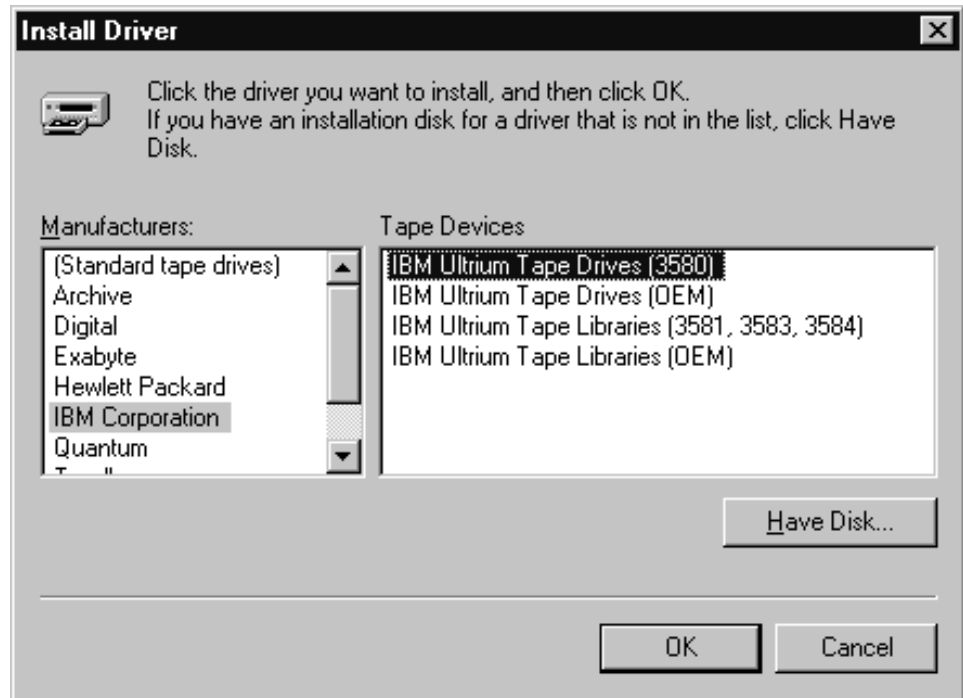


Figure 19. Install Driver Menu

9. If you installed the **Device Driver System Files** component in step 4 on page 227, Windows NT might ask the question shown in Figure 20. Click **Yes**, then skip to step 11. If the system prompts you for *ibmtape.sys*, browse to select the drivers directory (*c:\winnt\system32\drivers*). The system might also prompt you for *ntutil.exe*. Browse to the system32 directory (*c:\winnt\system32*).

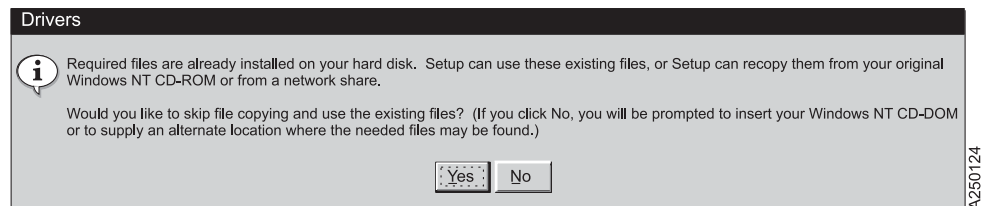


Figure 20. Windows NT Statement

10. If you **did not** install the **Device Driver System Files** component in step 4 on page 227, click **No** to the question in Figure 20. The operating system will prompt you for the location of the driver files.
11. Reboot the system for the operating system to start the drivers.

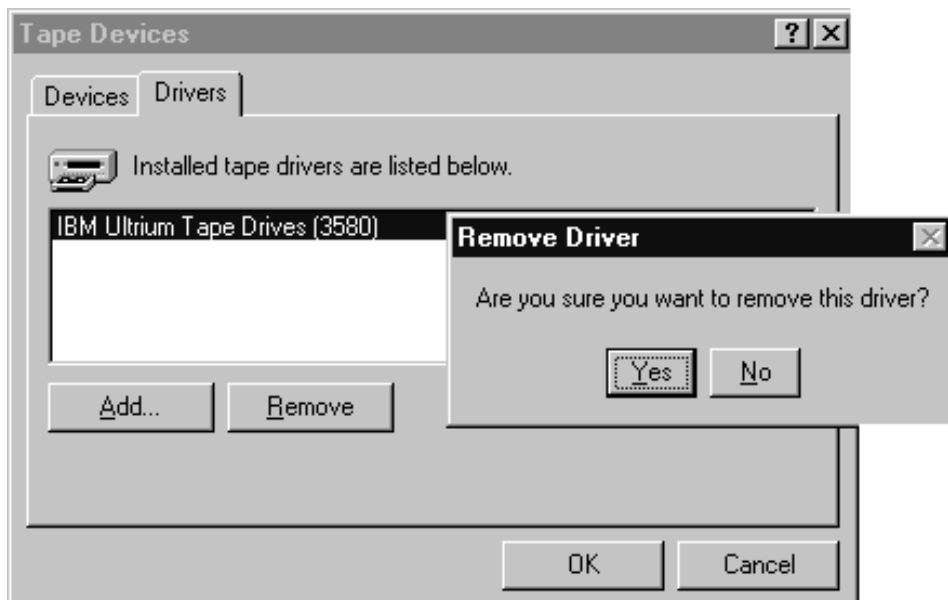
Removal Procedure

The Windows NT device driver and media mover device drivers manage all Ultrium devices and changers exclusively. If you use applications that have their own device drivers that access Ultrium devices and changers, you must remove the Windows NT device driver and media mover device drivers before installing and configuring those other applications.

1. Log on as Administrator.
2. Click **Start**, move to **Settings**, then click **Control Panel**.

Windows NT Device Driver (IBMtape)

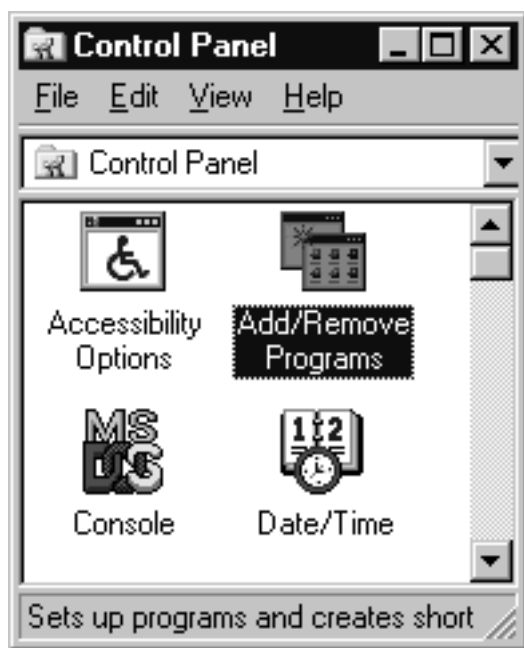
3. Double-click **Tape Devices**.
4. Click the **Drivers** tab.
5. Highlight **IBM Ultrium Device Driver**, then click **Remove**. When you see the message **Are you sure you want to remove this driver?**, click **Yes**. See Figure 21.



A250132

Figure 21. Remove Driver Menu

6. Click **Add/Remove Programs** in the **Control Panel**. See Figure 22.



A250126

Figure 22. Control Panel Selection

7. Highlight **IBM Ultrium Device Drivers**, then click **Add/Remove...**. See Figure 23.

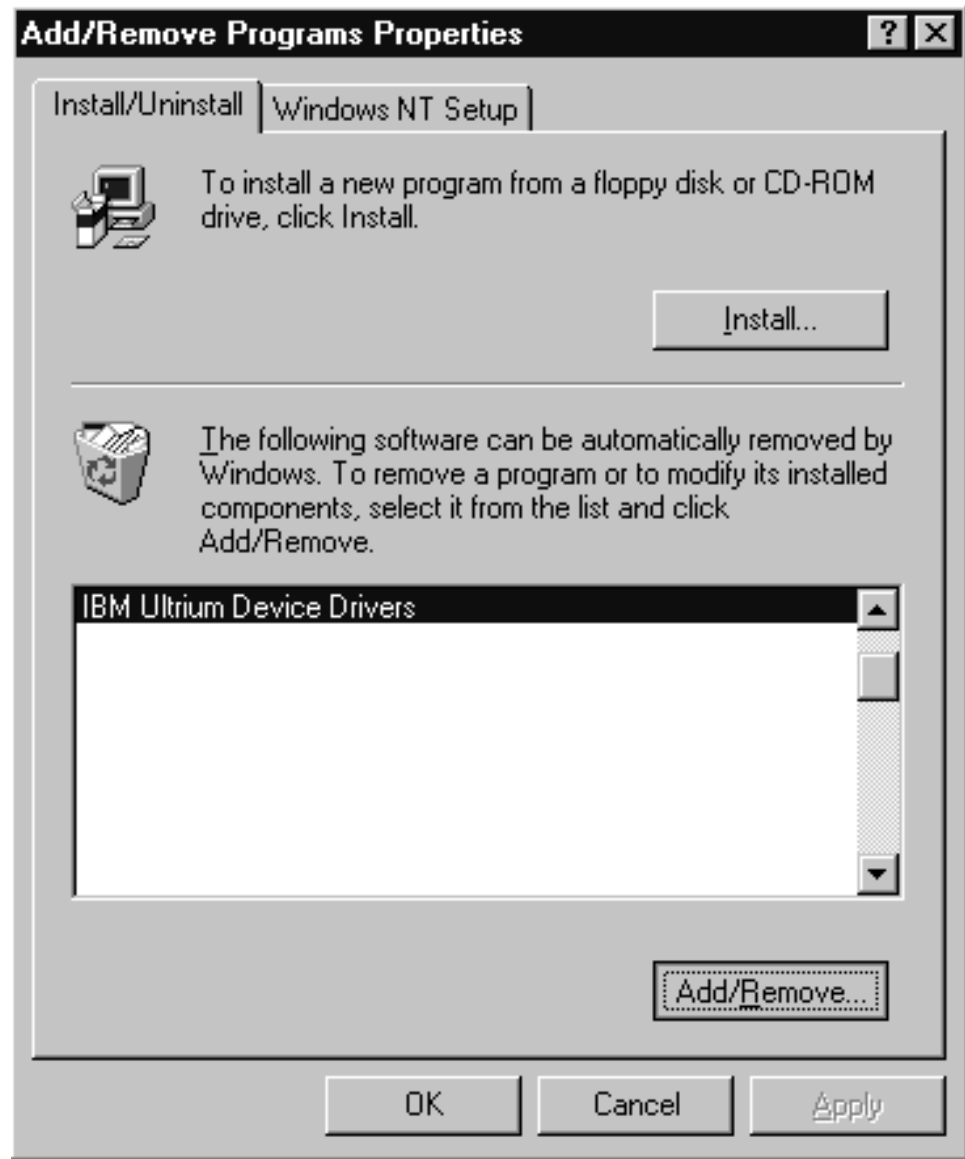


Figure 23. Add/Remove Properties

8. Select the **Remove** option, then follow the *InstallShield Wizard* to uninstall the drivers. See Figure 24 on page 236.



Figure 24. Drive Removal Menu

Note: Uninstalling the drivers removes all Ultrium device driver files as well as registry settings and other components, such as the *IBM Ultrium Device Drivers: Installation and User's Guide* and the Device Driver Depot.

9. Click Next >.
10. Shut down and reboot the system.

Manual Starting and Stopping Procedures

The IBM Ultrium device and changer drivers support being stopped and started without a reboot. If the Ultrium device and changer drivers are used in conjunction with other drivers that support being stopped and started without a reboot, you will be able to switch between device drivers without rebooting the system.

To control manually when either or both of the Ultrium device drivers start or stop, set the startup mode to **Manual**, then start the device driver manually when required.

To set the startup mode to **Manual**:

1. Log on as Administrator.
2. Click **Start**, move to **Settings**, then click **Control Panel**.
3. Double-click **Device**.
4. The changer driver and the device driver are named IBM Ultrium Changer Driver and IBM Ultrium Device Driver, respectively. Scroll down until they are visible in the window.
5. Select the driver, click **Startup...**, select **Manual**, then click **OK**. See Figure 25 on page 237.

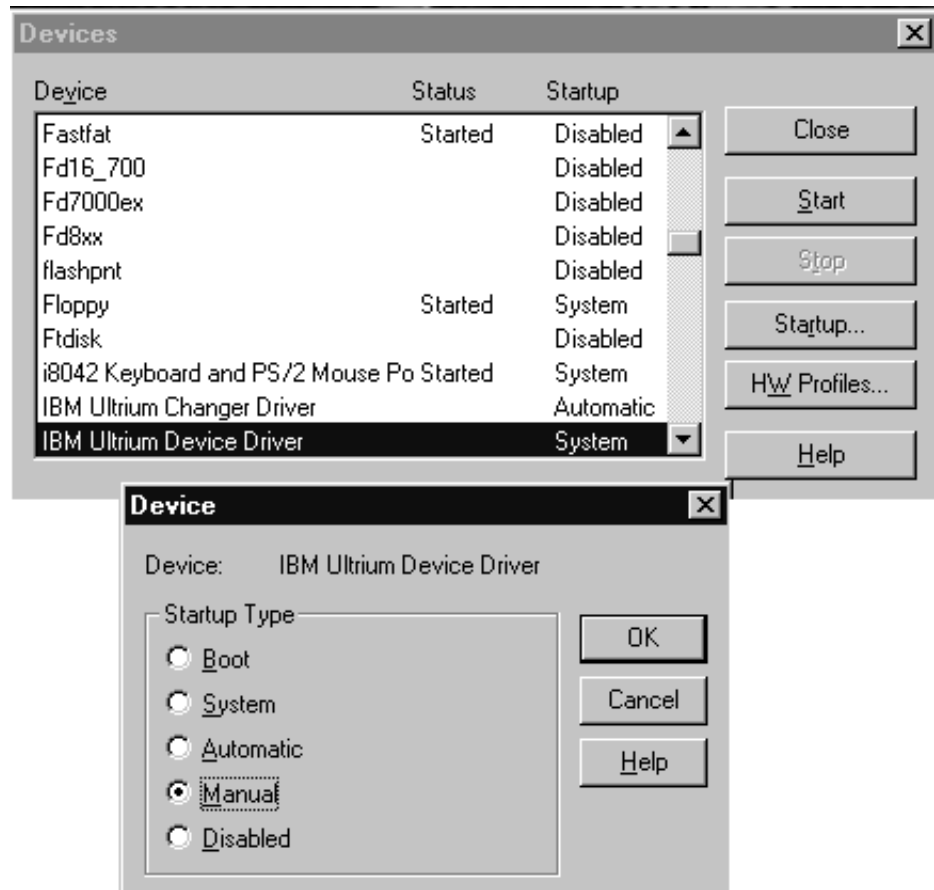


Figure 25. Manual Starting and Stopping Menu

6. If required for other drivers, repeat steps 4 and 5 of this procedure for each driver.

To start a driver manually:

1. Ensure that other drivers that could conflict are not started.
2. Perform steps 1 through 4 on page 236.
3. Select the driver, then click **Start**.
4. If necessary, repeat for the other driver.

To stop a driver manually:

1. Perform steps 1 through 4 on page 236.
2. Select the driver, then click **Stop**.

A250134

Windows NT Device Driver (IBMtape)

Chapter 42. Windows 2000 and Windows Server 2003 Device Driver Management

This chapter describes how to install, remove, and disable the Microsoft Windows 200x Tape Device Driver for the IBM Ultrium devices.

Installation Overview

The installation process consists of the following steps:

1. Verify that the hardware and software requirements have been met.
2. Install the host bus adapters and drivers.
3. Copy the Ultrium device driver files to your hard disk.
4. Shut down the system.
5. Connect the Ultrium devices to the host bus adapters.
6. Power On the Ultrium devices.
7. Set the Ultrium device addresses.
8. Reboot the system.
9. Log on as Administrator.
10. Install and configure the devices and device drivers using the Device Manager or the installation application.

All drives accessible from a medium changer must be on the same physical SCSI bus as the changer.

Installation Procedures

These procedures make the following assumptions:

- No other driver is installed that claims the Ultrium devices. If previous versions of the Ultrium device driver exist on your system that were installed with these procedures, uninstall them using the uninstall procedures in this documentation.
- If you have a previous version of the driver that was an InstallShield installation, see the Readme file for instructions on uninstalling that package.
- The host bus adapter is installed and configured properly and is running supported microcode and driver levels.
- Drivers are identified by the following conventions, where *nnnn* refers to a version of the driver. If there is more than one version, use the latest.
 - Windows 2000 and Windows Server 2003, 32-bit
IBMTape.W32_*nnnn*.zip
 - Windows Server 2003, 64-bit
IBMTape.W64_*nnnn*.zip

Installation Setup

To begin the installation procedures, perform the following steps:

1. Log on as Administrator.
2. Insert the Tape Device Drivers and SMI-S Agent CD in the CD-ROM drive or download the appropriate file from a subdirectory of

Windows 200x Device Driver (IBMTape)

`ftp://ftp.software.ibm.com/storage/devdvr/Windows/`. Drivers for Windows 2000 are in the Win2000 subdirectory. Drivers for Windows Server 2003 are in the Win2003 subdirectory. If you download the files, ensure that you use FTP "binary" mode. The most current driver will be labeled as IBMTape.Wxx_CURRENT.zip.

3. Unzip the driver package to a directory of your choice.

Note: If using the CD install method, you will find the appropriate driver package in the Drivers/Windows directory on the CD.

4. Shut down your system and ensure that the Ultrium devices are connected to your host bus adapter and configured properly.
5. Restart the system and log on as Administrator.

The **Found New Hardware Wizard** might start and indicate that tape devices were found. If you see the wizard, select Cancel and proceed to Step 6. If the **Found New Hardware Wizard** does not start, proceed to Step 6.

6. Right-click **My Computer** on your desktop and select Manage.
7. After a few seconds, a Computer Management console opens. See Figure 26.

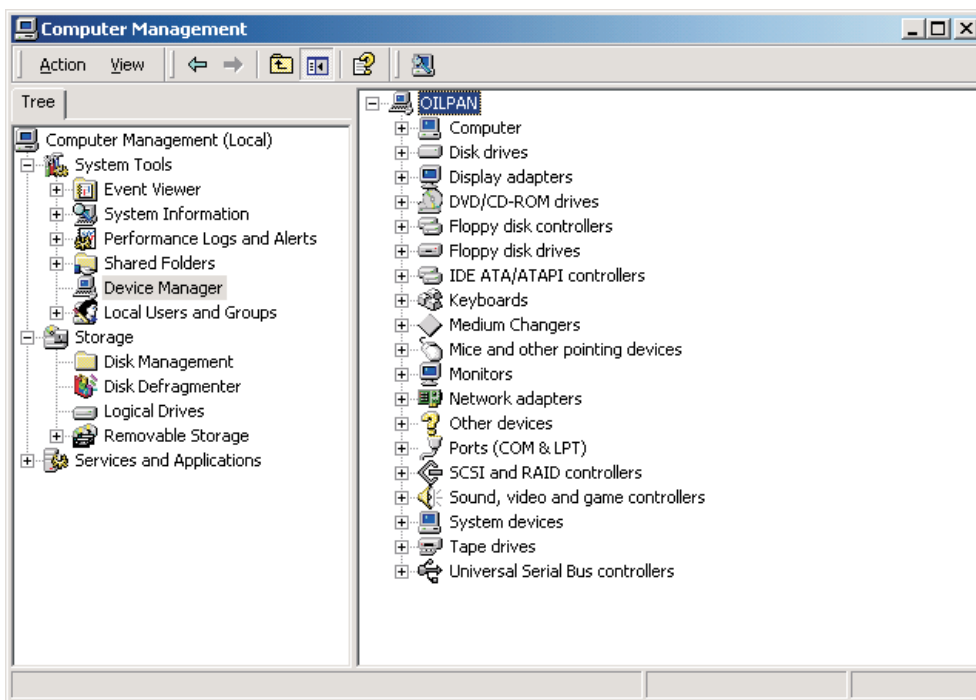


Figure 26. Computer Management Console

8. In the **Device Manager** tree, find the Ultrium devices you are installing. Tape devices might be located under the Tape drives node. Changer devices might be located under the Medium Changers node. Either device could be listed as an Unknown device or could appear under the Other devices node. See Figure 27 on page 241

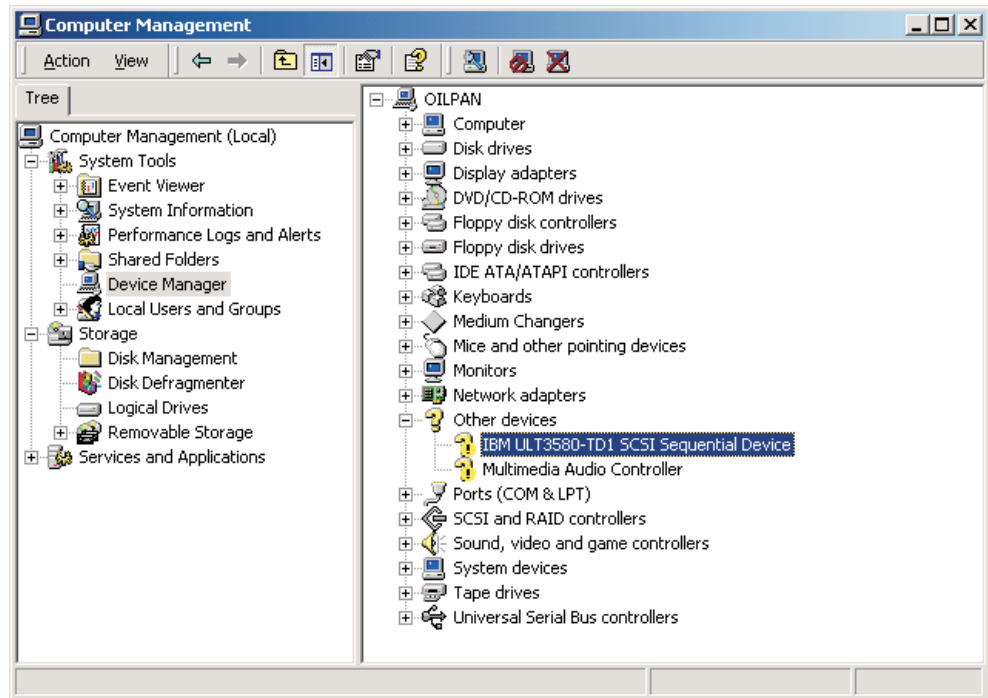


Figure 27. Computer Management Console with Ultrium Device Selected

Note: If you are installing the device driver for Medium Changers, see “Installing the Windows 200x Driver on Medium Changers” on page 251. If you are installing the device driver for Tape Devices, see “Installing the Windows 200x Driver for Tape Devices.”

Installing the Windows 200x Driver for Tape Devices

To install the device driver for Tape Devices on Windows 2000, do the following:

1. Right-click the Ultrium device and select **Properties**. A window opens.
2. Click the **Reinstall Driver** See Figure 28 on page 242. If there is no **Reinstall Driver ...** button on your screen, click on the **Driver** tab and then click **Update Driver**.

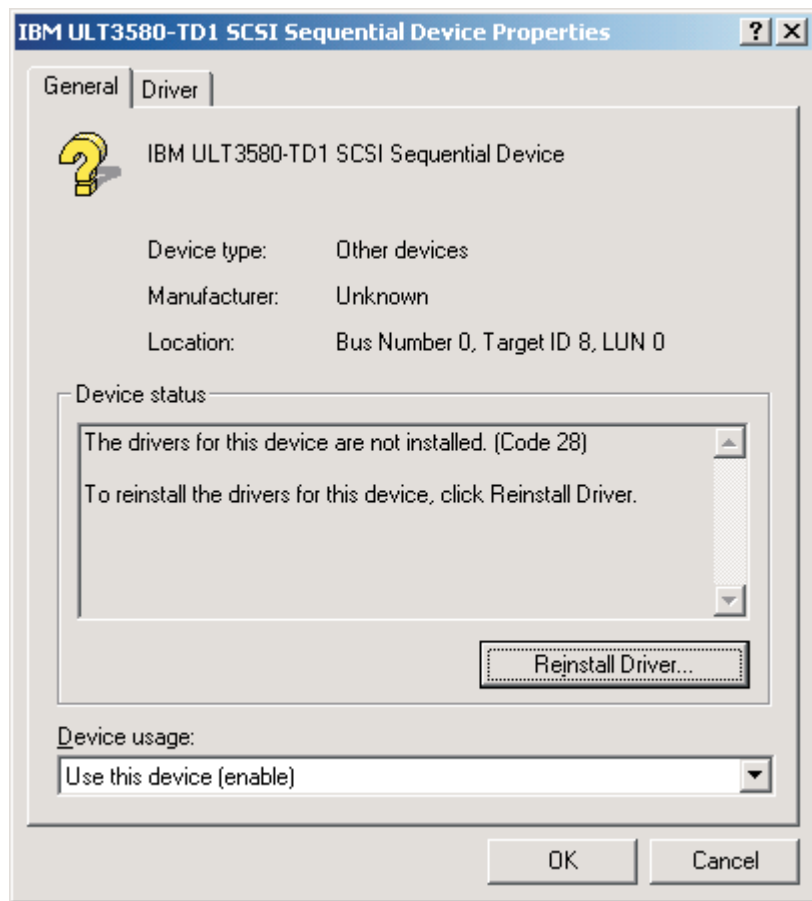


Figure 28. Device Driver Properties with Reinstall Driver ... Selected

3. If you are installing the Windows Server 2003 driver, see “Installing the Windows Server 2003 Driver for Tape Devices” on page 248. If you are installing the Windows 2000 driver, continue to step 4.
4. An **Upgrade Device Driver Wizard** window opens. Click **N**ext > to open the window shown in Figure 29 on page 243. Click **S**earch for a suitable driver for my device (recommended), then click **N**ext >. A **Locate Driver Files** screen appears.

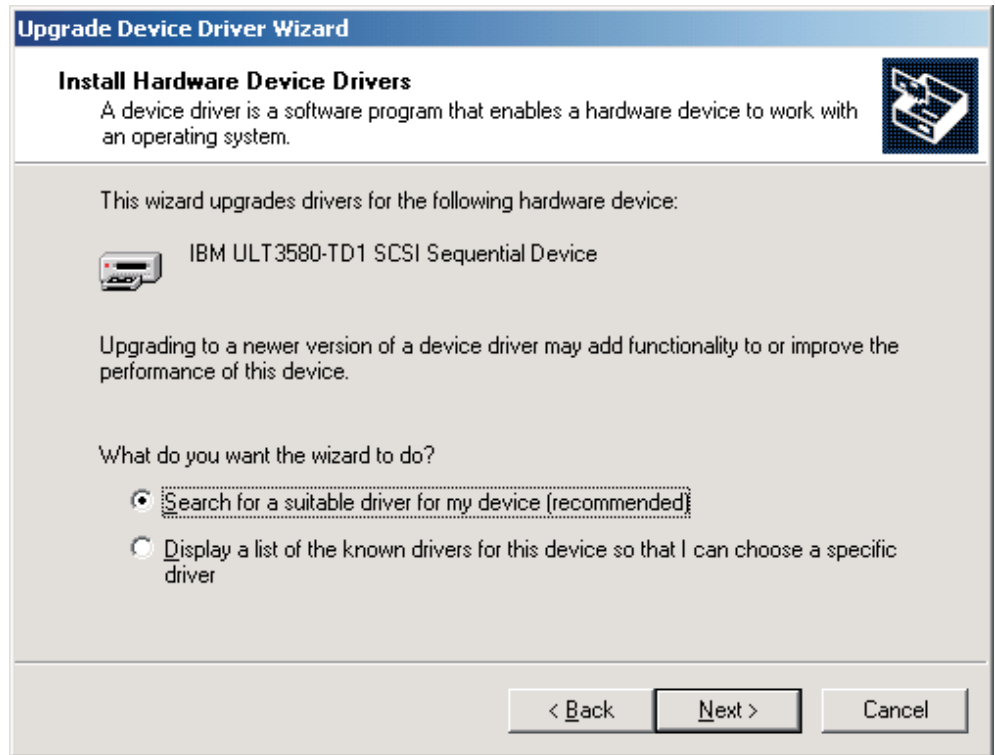


Figure 29. Install Hardware Device Drivers Dialog

5. Select the **Specify a location** checkbox and clear all other checkboxes. Click **Next >**. See Figure 30 on page 244.

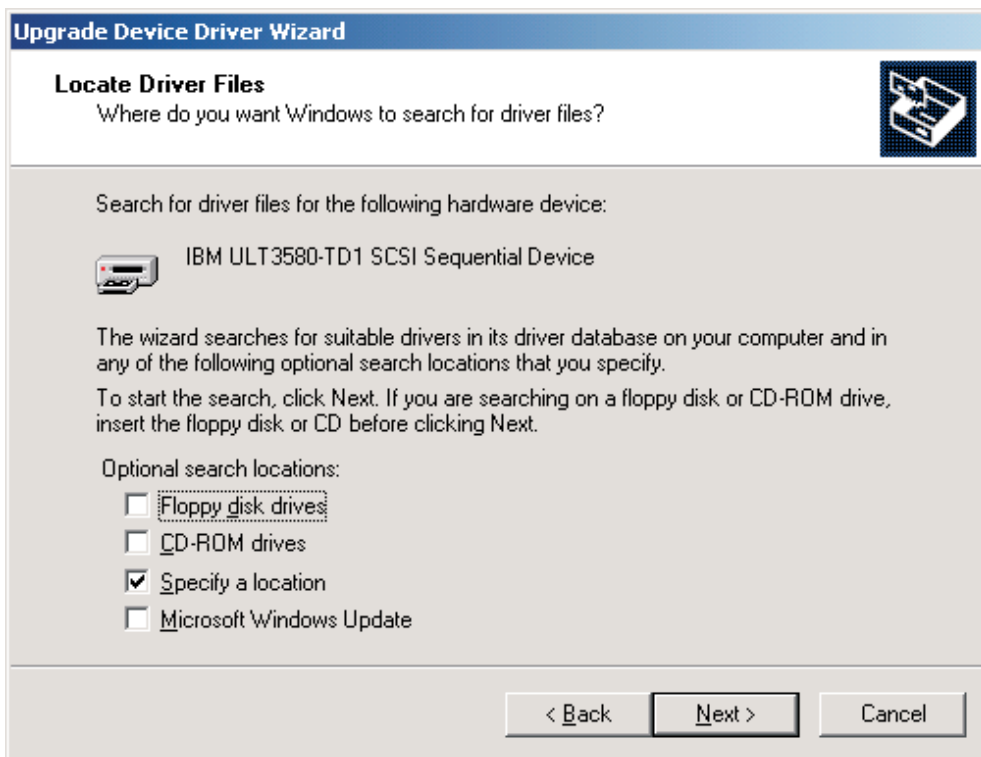


Figure 30. Locate Driver Files Screen with Specify a Location Selected

6. A screen appears with a prompt for the location of the device driver files. You should specify the directory into which you unzipped the driver files. For example, if you unzipped the files into d:\Driver Install, you would type d:\Driver Install in the edit. See Figure 31.

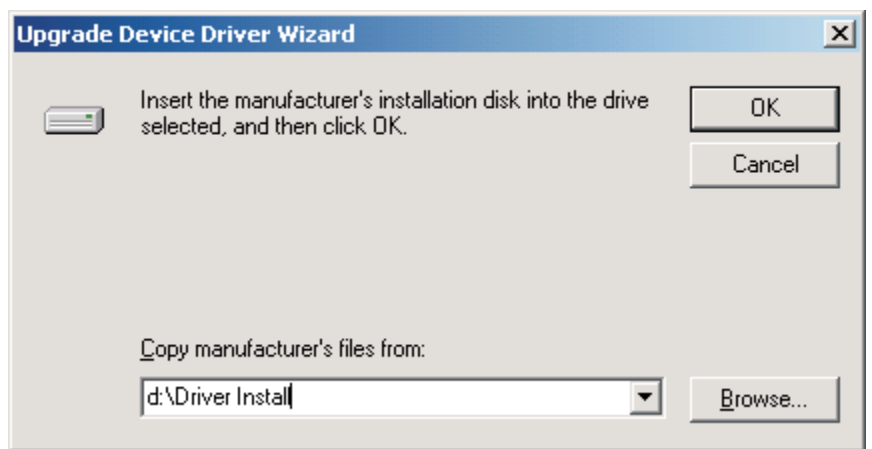


Figure 31. Specifying the Location of the Driver

This directory contains the most recent versions of the driver (with .sys extensions), the information files (with .inf extensions), and the utility and uninstall programs (with .exe extensions).

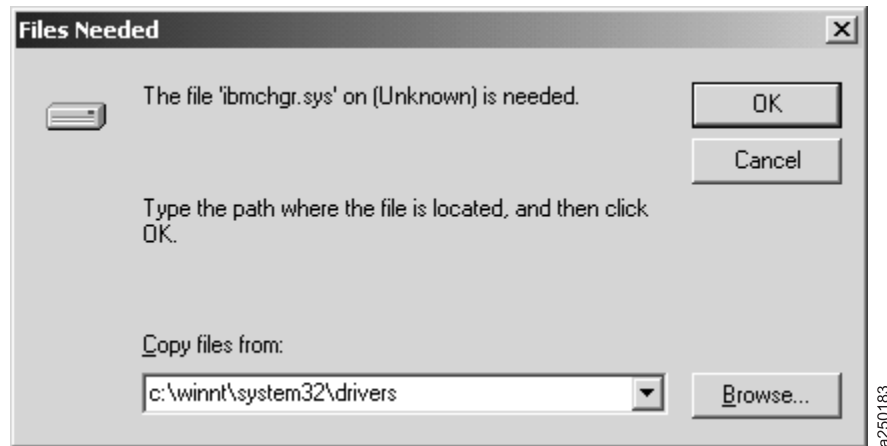


Figure 32. Files Needed Menu — ibmchgr.sys

7. Click **OK**.
8. After a brief moment of searching, the operating system should present the **Driver Files Search Results** screen. Figure 33 contains an example of a screen that will be presented when it finds an IBM Ultrium Generation 1 drive. Depending on the device you are installing, this and subsequent screens could have minor differences. Click **Next >**.

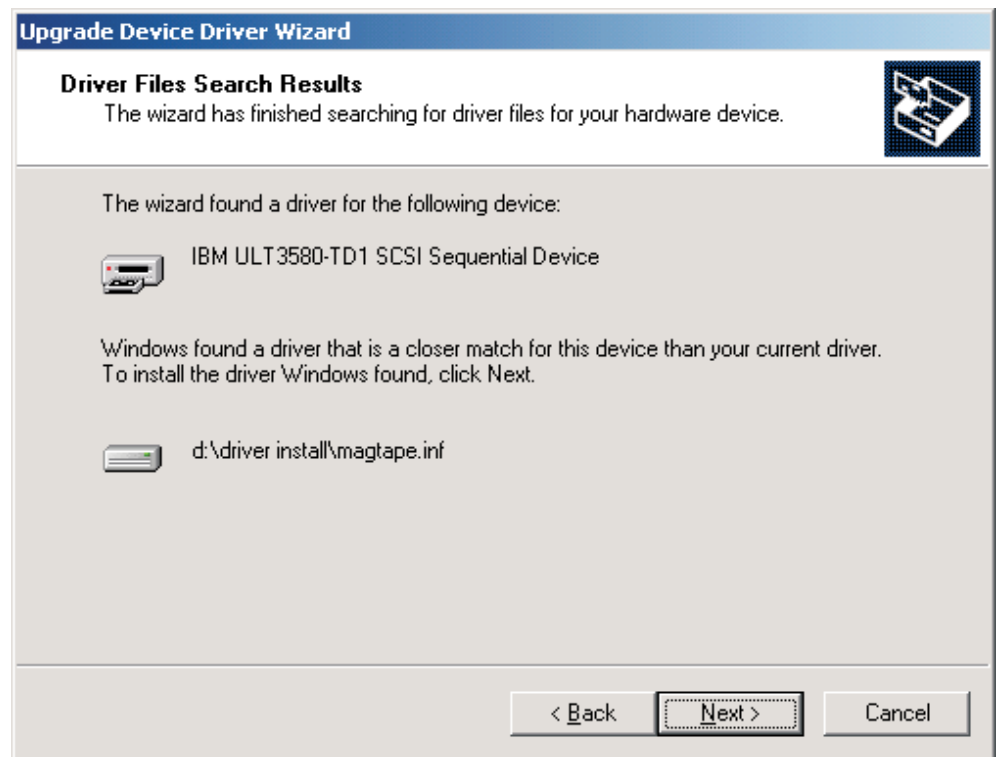


Figure 33. Driver Files Search Results

9. The **Driver Files Search Results** might also prompt you to select between two suitable drivers. See Figure 34 on page 246. If you see this screen, select the check box and click **Next**.

Windows 200x Device Driver (IBMtape)



Figure 34. Selecting Another Driver to Install

You will be presented with a Driver Files Found screen. See Figure 35.

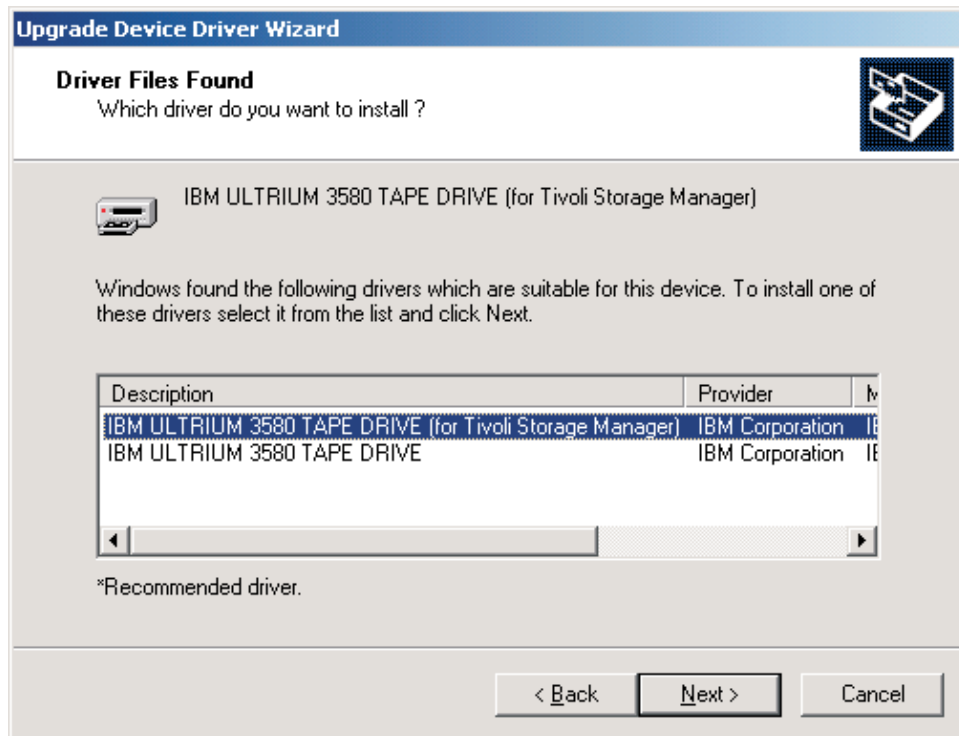


Figure 35. Driver Files Found screen

10. The drivers you can select from include a driver customized for Tivoli Storage Manager (TSM) and a generic driver used to run other applications, such as Microsoft Removable Storage Manager or Microsoft Backup. Consult the README file on the CD or FTP site to determine which driver to use. Select the driver that best suits your purposes and click **Next**.
11. If the installation was successful, you will be prompted to click **Finish** to complete the installation process. See Figure 36.

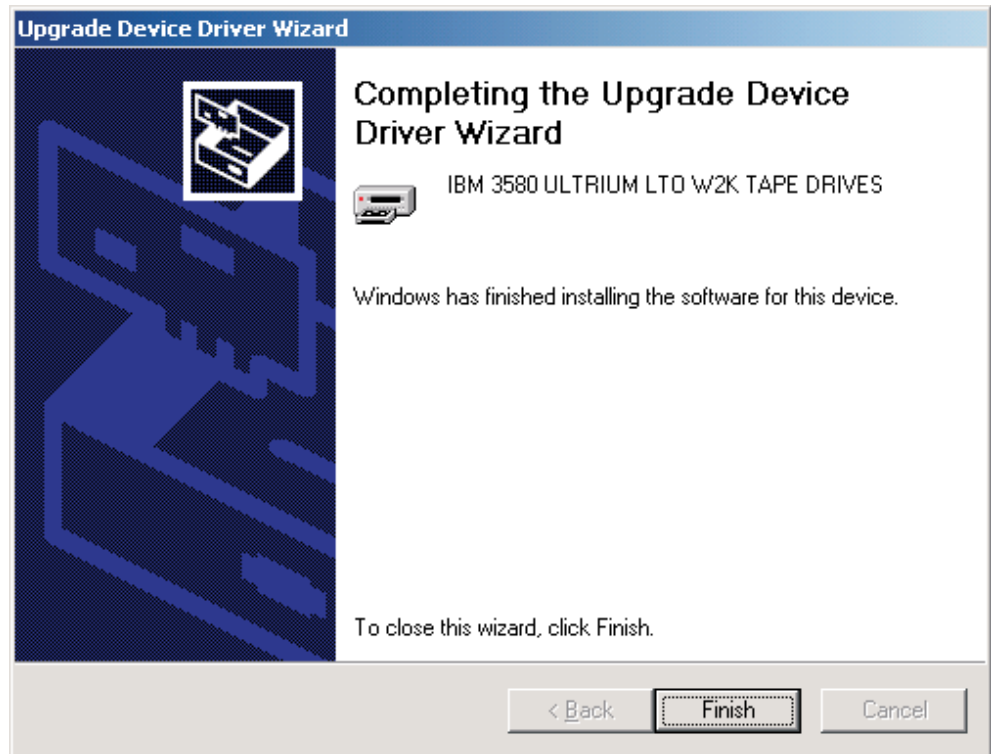


Figure 36. Completing the Upgrade Device Driver Wizard

12. You should be able to verify that the device was installed correctly. See Figure 37 on page 248. Repeat all steps in this section for every device you install. Remember, if you are installing a tape library you will need to install drivers for both the tape drives and the medium changer. To install the medium changer driver, skip to "Installing the Windows 200x Driver on Medium Changers" on page 251.

Windows 200x Device Driver (IBMtape)

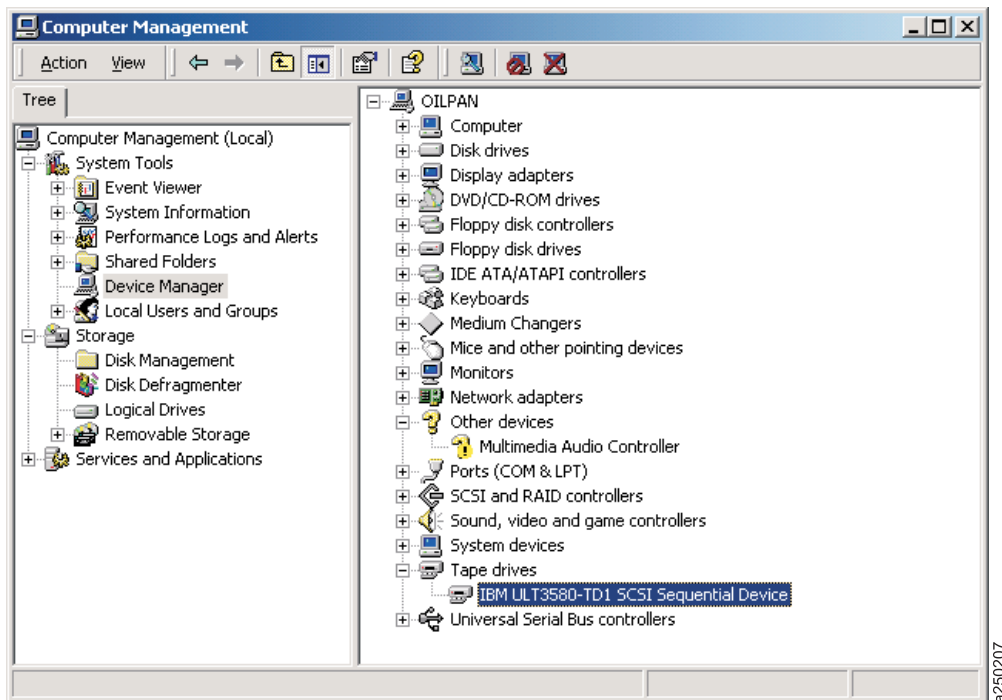


Figure 37. Verifying the Ultrium Device Was Installed Correctly

Note: To verify that an IBM Ultrium device is working properly, you can follow the instructions in Appendix B. the Microsoft Windows System Tape Device Attachment Test.

Installing the Windows Server 2003 Driver for Tape Devices

Perform the following steps to install the Windows Server 2003 driver on the discovered Tape devices. For Medium Changers, see “Installing the Windows 200x Driver on Medium Changers” on page 251.

1. A Hardware Update Wizard opens. Select **Install from a list or specific location (Advanced)** and click **Next >**. See Figure 38 on page 249.



Figure 38. Hardware Update Wizard Welcome Screen

2. The Wizard will prompt you to specify your search and installation options. Select the **Search for the best driver in these locations** radio button. Deselect the **Search removable media (floppy, CD-ROM...)** check box and select the **Include this location in the search** check box. Type the name of the directory into which you unzipped the driver files. See Figure 39.

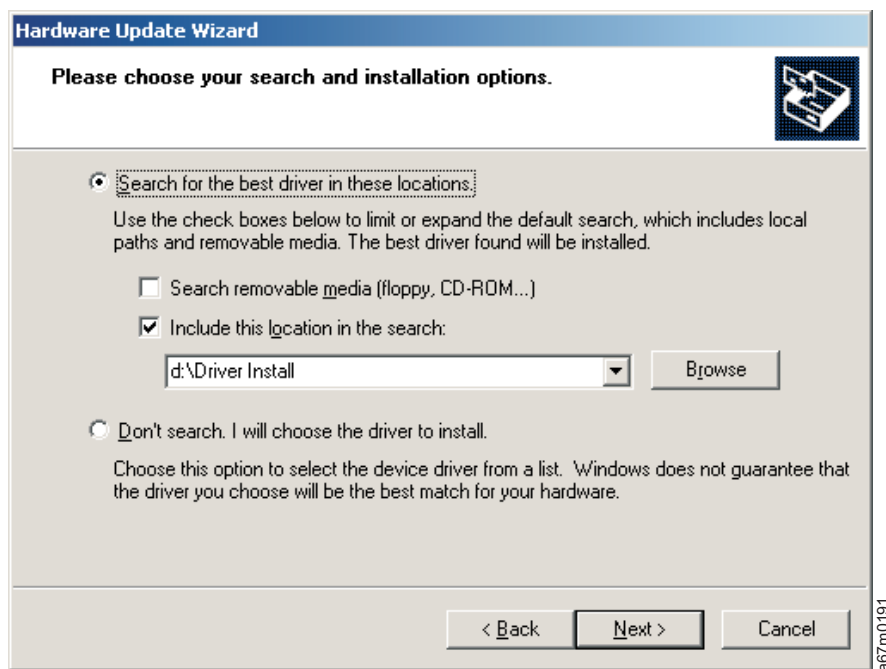


Figure 39. Search and Installation Options

Windows 200x Device Driver (IBMtape)

3. If you are installing a driver that has not been certified by the Microsoft Windows Hardware Quality Laboratories (WHQL), you will be presented with a warning screen. See Figure 40. If you want to continue installing the driver, select **C**ontinue Anyway.

Note: All drivers released by IBM have been through a complete test to ensure that they are stable and conform to specified requirements.



Figure 40. Windows Logo Testing Screen

4. You will be prompted to complete the installation. Click **F**inish. See Figure 41 on page 251.

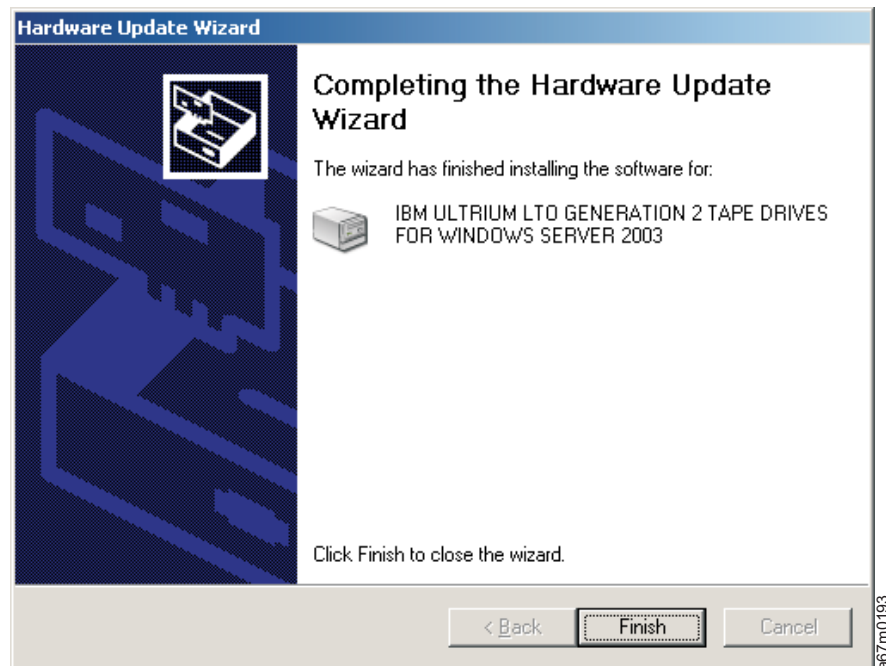


Figure 41. Completing the Hardware Update Wizard

Note: To verify that an IBM Ultrium device is working properly, follow the instructions in Appendix B, the Microsoft Windows System Tape Device Attachment test.

5. Repeat the steps provided in “Installing the Windows 200x Driver for Tape Devices” on page 241 for every Tape device you install. Remember, if you are installing a tape library you will need to install drivers for both the tape drives and the medium changer. To install the medium changer driver, proceed to “Installing the Windows 200x Driver on Medium Changers.”

Installing the Windows 200x Driver on Medium Changers

To install the device driver for all discovered Medium Changers on your system, you will need to run the installation application provided with the driver files.

1. Open Windows Explorer and locate the directory into which you unzipped the driver files, for example, d:\Driver Install. See Figure 42 on page 252.

Windows 200x Device Driver (IBMtape)

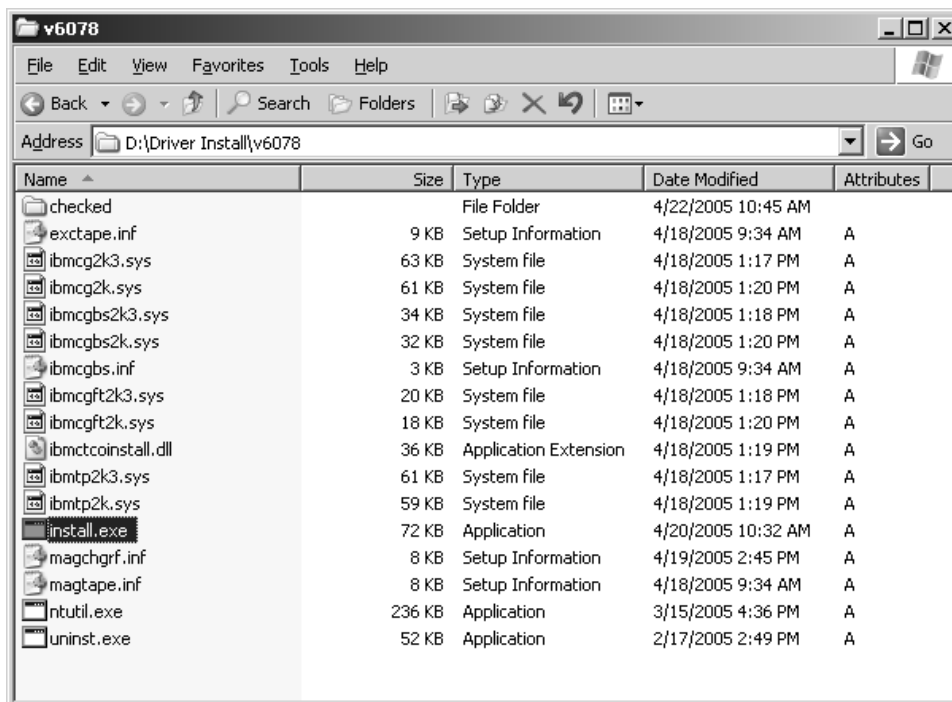


Figure 42. Installation Application in Windows Explorer

2. Double-click the **install.exe** file. The installation application will begin and automatically install the device drivers for the Medium Changers. This application applies to both the Windows 2000 and Windows Server 2003 drivers.
3. When the installation is complete, click **OK**.

Note: To verify that an IBM Ultrium device is working properly, follow the instructions in Appendix B, the Microsoft Windows System Tape Device Attachment test.

Device Removal or Disable Procedure

If you need to remove a device, or if you are altering the hardware configuration, you should uninstall or disable the device first.

1. Right-click **My Computer**, select **Manage** to open the Computer Management Console, and click **Device Manager**.
2. Right-click the Ultrium device you wish to uninstall and select **Uninstall** If you want to disable the device without uninstalling it, you can select **Disable**.
3. You will be prompted to confirm the uninstallation. Click **OK**.

Note: This removal procedure will remove the device from the device tree, but it will not uninstall the device driver files from your hard disk.

Uninstalling the Device Drivers

To uninstall the device drivers from the system, which includes deleting the system files and deallocating other system resources, run `uninst.exe` from the zip archive. This removes all the files in the system directories that were created during the installation of the device driver. It does not delete the .zip file or the files that were extracted from the .zip file. If you desire to remove these files, you will need to delete them manually.

Windows 200x Device Driver (IBMtape)

Chapter 43. Alternate Pathing Support for Library Control Path Failover

Note: The library control path failover feature code must be installed prior to enabling the alternate pathing support in the Windows 200x device drivers. Refer to Table 2 on page 12 to determine which feature code is required for your machine type.

The Windows device driver alternate pathing support will configure multiple physical control paths to the same logical library within the device driver and provide automatic failover to an alternate control path when a permanent error occurs on one path. This is transparent to the running application.

For example, consider a simple multipath connection that consists of two Host Bus Adapters (HBA) that are connected to a library with two drive control ports enabled. The first HBA is connected to the first control port drive, and the second HBA is connected to the second control port drive. This connection provides two physical control paths to the same library for redundancy if one path from an HBA to the library fails.

When the tape library is connected to the Windows host by two or more control paths, Windows will enumerate each path and all paths will appear in Device Manager as separate libraries. Each "logical library" may represent a physical path to the same library. Failover will be coordinated by a separate driver, called the Changer Bus Enumerator. When a library appears in Device Manager, the driver for the library will send notification messages to the Changer Bus Enumerator. The Changer Bus Enumerator will determine whether the library path represents a new, unique physical library or if it is another path to a library that already exists on the bus. The Changer Bus Enumerator will ensure that all paths to one library can be accessed through the same device handle (for example, \\.\Changer0).

Without the IBMtape alternate pathing support, if an application opens \\.\Changer0 and a permanent path error occurs (because of an HBA, cable, or drive control port failure, for example), the application fails. It is possible to initiate manual failover by restarting the application on the alternate logical device (which will have a unique name, such as \\.\Changer1), but the failing transaction has to be restarted from the beginning.

When the alternate pathing support is enabled, the device driver configures each library internally as a single device with multiple paths. The application can still open and use only one logical device at a time. If an application opens \\.\Changer0 and a permanent path error occurs, the device driver initiates failover error recovery automatically on the alternate path to \\.\Changer0. If successful, the current operation continues on the alternate path without interrupting the application.

Enabling and Disabling Alternate Pathing Support

Alternate pathing support is enabled automatically by default when the Windows 200x medium changer device driver is installed on a Windows system. The Windows 200x device driver provides an executable application for you to enable or disable the library control path failover (CPF) support. To enable the CPF support for all of the paths, no action is required. To disable the CPF support for all of the paths, follow these steps:

1. Open Windows Explorer and locate the directory into which you unzipped the driver files.
2. Inside the reg directory, you will find two files. Double click the DisableCPF.reg file.
3. Reboot your system. The CPF support will be disabled.
4. To re-enable CPF support, repeat steps 1-3, but double click on the EnableCPF.reg file.

Querying Primary and Alternate Path Configuration

When the device driver configures a logical device with alternate pathing support enabled, the first device configured always becomes the primary path. When a second or more logical device is configured with alternate pathing support enabled for the same physical device, it configures as an alternate path. The device driver supports an arbitrarily large number of libraries and library paths.

The primary and alternate path information can be obtained by use of the DeviceIoControl function with the CPF_IOCTL_BUS_QUERY_DEVICES control code. See "Chapter 20. Windows 2000 and Windows Server 2003 Programming Interface" in the *IBM Ultrium Device Drivers: Programming Reference*, for this IOCTL definition.

Chapter 44. Windows Utility Program (ntutil)

The Windows Utility Program (*ntutil*) allows easy operation of your Ultrium devices and changers. It is supported only with the IBM-supplied device drivers.

You can use the *ntutil* program for the following purposes:

- Help determine if there is a problem with hardware or connections
- Determine which devices are recognized by the device and changer drivers
- Force a driver dump
- Load new microcode
- Send SCSI commands to the hardware
- Receive the status of SCSI commands
- Obtain sense data for SCSI commands that encounter errors

The tool can be run in interactive mode or batch mode. When it is in interactive mode, a menu is presented to the customer and the customer issues one command at a time and is presented with the results of that command. In batch mode, the customer uses an editor to create a file that contains *ntutil* commands, which are presented to the tool one by one.

Calling ntutil

The tool can be called as a command from the command-line or from within a shell script:

```
ntutil <-f input-file> <-o output-file> <-t tape-path-special-file-name>  
      <-c changer-path-special-file-name> <-l >  
      <-d >
```

The options are:

-f input-file

Specifies the input file for batch mode.
If a file is specified, NTUTIL will execute in batch mode and read input from this file.

-o output-file

Specifies the output file for batch mode.
The default for this file is NTUTIL.OUT.

-t tape-special-file-name

Specifies the tape device special file value (for example, `tape0`) to substitute when executing an open (for both batch and interactive mode).

-c changer-special-file-name

Specifies the changer device special file value (for example, `1b0.1.2.3` for Windows NT, `Changer0` for Windows 200x) to substitute when executing an open (for both batch and interactive mode). The special value `def_lun_1` specifies that an open uses the default lun 1 associated with the `tape-special-file-name`.

Windows Utility Program (ntutil)

- l
Specifies that an open will open both the tape path special file and the changer path special file (for both batch and interactive mode).
- d
Turns on internal tracing printouts in the output. Used only for tool debugging.
- b
Begins ntutil in batch mode (same as -f, but the default input file ntutil.in is used).

Note: If no parameters are specified, *ntutil* will operate in interactive mode.

Interactive Mode

When *ntutil* is called without the *-f* flag, it defaults to the Interactive (or Manual) mode. This mode allows a developer to interactively determine the kind of testing to be done. When in Interactive mode, *ntutil* provides a menu of functions that can be performed.

For a description of interactive modes, see “Batch Mode” on page 260.

The two modes are:

- Base mode (LUN0) commands, such as *open*, *close*, *read*, or *write*
- Library mode (LUN0 and LUN1), which supports *open*, *close*, *read*, and *write*, plus media mover commands such as *read element status* and *move media*.

Note: On Windows 200x platforms, library functions are available if the Removable Storage Management component of Windows 200x is stopped.

Base mode (only LUN0 of the device specified by the tape-path-special-file-name, for example, *tape0*, is accessed) is shown in Figure 43 on page 259.

Library mode (LUN0 specified by the tape-path-special-file-name, for example, *tape0*, and LUN1 specified by the changer-path-special-file-name, for example, *lb0.1.0.0*, are accessed) is shown in Figure 44 on page 260.

To issue SCSI commands, the device must be open. Open a SCSI device by issuing option 20 (*Open*). The device names can be obtained from command 88 (*Find Devices*). Tape devices names are of the format *tapen*, where *n* is a digit, 0, 1, and so on. If the device driver is stopped, then started without a reboot, the name will not be the same as it was previously. Rather it will be the next unused name in the operating system. For example, if there is one tape device defined on the Windows system, that device will be named *tape0* when the device driver is started the first time. If the device driver is stopped, then restarted, the name will be *tape1*. This behavior continues until the system is rebooted.

```

Test tool version x.x.x.x
Variable settings
===== BASE MODE =====
tape-special-file-name: tape0
gp->fd0=-1 gp->fd1=-1 block size=1024 block count=1
hex block id = 0000000000000000
return_error_when_fail 1 exit_on_unexpected_result 0 trace_flag 0

manual test menu:
=====
 1: set device special file          2: display symbols
 3: set block size R/W (now !0 fixed)
 5: set return error when fail       6: set/reset trace
 7: set exit on unexpected result    8: Library Mode
=====
20: open                             21: close
22: read                             23: write
24: read and display block           25: flush (buffer->media)
26: read block id                   27: erase
28: locate block id                 29: display block data
=====
30: write filemark(s)               31: rewind
32: forward space filemark(s)       33: unload
34: reverse space filemark(s)       35: load
36: forward space record(s)         37: return error
38: reverse space record(s)         39: test unit ready
43: set media parms (block size)    44: set dev parms (compression)
46: get device information           47: restore data
48: get medium information           49: inquiry
50: poll registered devices
53: space EOD
=====
70: system command
=====
80: Force Dump                      81: Read Dump
82: Update MicroCode                 83: Log Sense
84: Get Last Sense                   85: Get Version
                                     87: Read/Write Test

88: List registered devices
=====
99: return to main menu
=====
enter selection:

```

Figure 43. Base Mode

Windows Utility Program (ntutil)

```
Test tool version x.x.x.x
Variable settings
===== LIBRARY MODE =====
tape-special-file-name: tape0, changer-special-file-name: lb0.1.2.3
gp->fd0=-1 gp->fd1=-1 block size=1024 block count=1
hex block id = 0000000000000000
return_error_when_fail 1 exit_on_unexpected_result 0 trace_flag 0

manual test menu:
=====
 1: set device special file          2: display symbols
 3: set block size R/W (now !0 fixed)
 5: set return error when fail       6: set/reset trace
 7: set exit on unexpected result     8: Base Mode
=====
10: ioctl return library inventory   11: ioctl move medium
12: ioctl initialize element status  13: ioctl get changer parameters
=====
20: open                             21: close
22: read                             23: write
24: read and display block           25: flush (buffer->media)
26: read block id                   27: erase
28: locate block id                 29: display block data
=====
30: write filemark(s)               31: rewind
32: forward space filemark(s)       33: unload
34: reverse space filemark(s)       35: load
36: forward space record(s)         37: return error
38: reverse space record(s)         39: test unit ready
43: set media parms (block size)    44: set dev parms (compression)
46: get device information           47: restore data
48: get medium information           49: inquiry
50: poll registered devices
53: space EOD
=====
70: system command
=====
80: Force Dump                      81: Read Dump
82: Update MicroCode                83: Log Sense
84: Get Last Sense                   85: Get Version
                                     87: Read/Write Test
88: List registered devices
=====
99: return to main menu
=====
enter selection:
```

Figure 44. Library Mode

Batch Mode

Batch input files can contain the following kinds of statements:

- comments
- command
- set
- type
- pause
- delay
- system
- symbols
- exit

Each type of statement is described in the following section.

Comments

Any line starting with a pound sign (#), any line starting with a space, or any blank line is considered a comment and is ignored.

Command Statements

Device driver function is exercised by command entries in the input file. Command statements must be on a single line of the input file. The command and command text are case sensitive, but leading or embedded blanks are ignored.

```
command command-text <result-text>
```

This statement is used to execute a tape command and to test the command completion status for an expected result.

Supported Command Text Fields

The command text is used to specify the tape operation to perform. The following section describes the possible values for this field.

close

This command calls the device driver CloseHandle entry point and closes the tape device special file opened previously. The RC is always 1.

SYNTAX: close

```
for example, close
```

FUNCTION Tested:

device_info

This command calls the device driver GetTapeParameters entry point with an operation of GET_TAPE_DRIVE_INFORMATION. If the command is successful, the information returned is in a TAPE_GET_DRIVE_PARAMETERS structure, described in the Microsoft Software Developers Kit (SDK). The information will be formatted and printed. This includes information such as ECC, compression, default block size, and features.

SYNTAX: device_info

```
for example, device_info
```

FUNCTION Tested:

display_block (data)

This command displays the data buffer contents for the last read or write transfer.

SYNTAX: display_block

```
for example, display_block
```

FUNCTION Tested:

dump_blocks (read and display blocks)

This command calls the device driver ReadFile entry point and attempts to read from the tape device special file opened previously. The amount of data to be read will depend of the current settings of the *block_size* and *block_count* variables and on the number of records specified. The first 16 bytes of each record will be printed.

Windows Utility Program (ntutil)

This command transfers **n** records. A record will have a length of `block_size*block_count` for fixed block mode and `block_count` for variable mode. For example, if `block_size=1024`, `block_count=2` and `records=3`, then three 2048 transfers will be done. If the operation succeeds, the RC will show the total number of bytes transferred.

SYNTAX: `dump_blocks`
`dump_blocks records = n`

for example, `dump_blocks records = 2`

FUNCTION Tested:

erase

This command calls the device driver `EraseTape` entry point with an operation of `TAPE_ERASE_LONG`. This attempts to erase, then leave at load point the tape device special file opened previously.

SYNTAX: `erase`

for example, `erase`

FUNCTION Tested:

find_devices

This command (which corresponds to the *List registered devices* command on the interactive menu) searches the following registry key and looks for Ultrium identifiers:

```
"HARDWARE\DEVICEMAP\Scsi\Scsi Port W\Scsi Bus X\Target Id Y\Logical Unit Id Z"
```

It then prints a list of the SCSI devices supported by the IBM Ultrium device drivers.

SYNTAX: `find_devices`

for example, `find_devices`

FUNCTION Tested:

flush (buffer to media)

This command calls the device driver `WriteTapemark` entry point with an operation of `TAPE_FILEMARKS` and a count of 0. This writes the data buffer contents to the tape media of the tape device special file opened previously.

SYNTAX: `flush`

for example, `flush`

FUNCTION Tested:

force_dump

This command calls the device driver `DeviceIoControl` entry point with an operation of `IOCTL_SCSI_PASS_THROUGH` and a `CDB[0]` of `SCSIOP_SEND_DIAGNOSTIC`. This forces a microcode dump.

SYNTAX: `force_dump`

for example, `force_dump`

FUNCTION Tested:

forward_filemark

This command calls the device driver SetTapePosition entry point with an operation of TAPE_SPACE_FILEMARKS and a count of **n**. This attempts to forward space **n** files on the cartridge in the tape device special file opened previously.

SYNTAX: forward_filemark n

for example, forward_filemark 2

FUNCTION Tested:

forward_record

The forward_record command calls the device driver SetTapePosition entry point with an operation of TAPE_SPACE_RELATIVE_BLOCKS and a count of **n**. This attempts to forward space **n** records on the cartridge in the tape device special file opened previously.

SYNTAX: forward_record n

for example, forward_record 2

FUNCTION Tested:

get_last_sense

This command calls the device driver DeviceIoControl entry point with an operation of OBTAIN_SENSE. This displays the last sense data returned by a sense command.

SYNTAX: get_last_sense

for example, get_last_sense

FUNCTION Tested:

get_version

This command calls the device driver DeviceIoControl entry point with an operation of OBTAIN_VERSION. This displays the Ultrium Device Drivers Version Id string.

SYNTAX: get_version

for example, get_version

FUNCTION Tested:

init_element_status

This command calls the device driver DeviceIoControl entry point with an opcode of LIBRARY_AUDIT that will issue an Initialize Element Status command to the device.

SYNTAX: init_element_status

for example, init_element_status

FUNCTION Tested:

inquiry

This command calls the device driver DeviceIoControl entry point with an operation of IOCTL_SCSI_PASS_THROUGH and a CDB of SCSIOP_INQUIRY. If the command is successful, the information returned will be displayed. Specify *n=0* to obtain inquiry information from the drive. Specify *n=1* to obtain inquiry information from the changer. Only Inquiry page 0 is supported.

Windows Utility Program (ntutil)

SYNTAX: inquiry n

for example, inquiry 0

FUNCTION Tested:

load

This command calls the device driver PrepareTape entry point with an operation of TAPE_LOAD. This attempts to load the tape media into the drive of the tape device special file opened previously.

SYNTAX: load

for example, load

FUNCTION Tested:

locate_block_id

This command calls the device driver SetTapePosition entry point with an operation of TAPE_LOGICAL_BLOCK. This attempts to position to the last *read_block_id* value for the cartridge in the tape device special file opened previously, or the block ID set by the *set_block_id* function.

SYNTAX: locate_block_id

for example, locate_block_id

FUNCTION Tested:

log_sense (page_code)

This command calls the device driver DeviceIoControl entry point with an operation of IOCTL_SCSI_PASS_THROUGH and a CDB[0] of SCSIOP_LOG_SENSE. This reads the log sense page specified by the page code and displays the data.

The *page_code* (xx) is two hex characters specifying the log sense page to be read.

SYNTAX: log_sense page_code = xx

for example, log_sense page_code = 00

FUNCTION Tested:

medium_info

This command calls the device driver GetTapeParameters entry point with an operation of GET_TAPE_MEDIA_INFORMATION. If the command is successful, the information returned is in the TAPE_GET_MEDIA_PARAMETERS structure described in the Microsoft SDK. The information is formatted and printed. The information returned is *block_size*, *partition count*, and *write protect*.

SYNTAX: medium_info

for example, medium_info

FUNCTION Tested:

move_medium

This command calls the device driver DeviceIoControl entry point with an opcode of MOVE_MEDIUM. This attempts to move a data cartridge from a source element location to a destination element location inside the library. If this command is successful, the information returned will be displayed. In Windows NT, the source and destination addresses are the decimal equivalents of the hex Element Addresses described in the appropriate hardware manuals. In Windows 200x, the

addresses are remapped such that the first element of each type is mapped to 0, the second element is mapped to 1, and so on. Issue a *return_lib_inventory_all* command to obtain the mapping.

SYNTAX - Windows NT: `move_medium saddr = n daddr = n`

where saddr = decimal source address (moving from)
 daddr = decimal destination address (moving to)

for example, `move_medium saddr = 11 daddr = 13`

SYNTAX - Windows 200x: `move_medium stype = n saddr = n dtype = n daddr = n`
 where

stype = decimal source address type
 2=SE(ChangerSlot), 3=IEE (ChangerIEPort), 4=DTE (ChangerDrive)
 saddr = decimal source address (moving from)
 dtype = decimal target address type (see stype for supported values)
 daddr = decimal destination address (moving to)

for example, `move_medium stype = 2 saddr = 1 dtype = 4 daddr = 0`

FUNCTION Tested:

open

This command calls the device driver CreateFile entry point and attempts to open a tape device special file for LUN0 and LUN1.

SYNTAX: `open <tape-special-file-name> <changer-special-file-name> RW RO`

where RW means read/write
 RO means read only

The tape-special-file-name is in the form of `tapen`, and `n` is a numeric value from `0..x`, the value is assigned by the device driver during the boot process and can be found using the interactive modes option `50 poll` registered devices. This value may be specified explicitly in the open statement or overridden using the command-line `-t` option.

The changer-special-file-name, on Microsoft Windows NT 4.0, is in the form of `lbw.x.y.z`, `w` is the device SCSI ID, `x` is the changer device LUN (usually 1), `y` is the adapter SCSI bus (usually 0), and `z` is the port number where the SCSI adapter resides. These values are assigned during the boot process and can be found using the interactive modes option `50 poll` registered devices. This may also contain the value `def_lun_1` in which case the utility will open LUN 1 associated with the tape-special-file-name. This value may be specified explicitly in the open statement or overridden using the command-line `-c` option. If this value is specified, the tape-special-file-name must be specified.

The changer-special-file-name, on Microsoft Windows 200x, is in the form of `Changern`, and `n` is a numeric value from `0..x`, the value is assigned by the device driver during the boot process and can be found using the interactive modes option `50 poll` registered devices. This value may be specified explicitly in the open statement or overridden using the command-line `-t` option. You must stop the Removable Storage Manager (RSM) before opening the changer device on Windows 200x.

for example,

`open RW` means open the default tape-special-file-name
 and changer-special-file-name if in library
 mode or those special file names specified
 by the `-t` and `-c` options

`open tapex RW` means open `tapex` and if in library mode

Windows Utility Program (ntutil)

the default changer-special-file-name.
The -t and -c options will override tapex
and lbw.x.y.z. The mode will be RW.

```
open tapex lbw.x.y.z RW
means open tapex and if in library mode
the changer special file lbw.x.y.z.
The -t and -c options will override tapex
and lbw.x.y.z. The mode will be RW.
```

FUNCTION Tested:

poll_devices

This command searches the following registry key:

HARDWARE\DEVICEMAP\Scsi\Scsi Port W\Scsi Bus X\Target Id Y\Logical Unit Id Z for devices supported by the IBM Ultrium drivers. This command then attempts to open each of the devices found and issue INQUIRY and TEST UNIT READY SCSI commands to each open device. The status of all the devices will be printed on the screen. Executing this command will close all currently open devices. If another application is using the device, this command will not be able to query the status of that device.

SYNTAX: poll_devices

for example, poll_devices

FUNCTION Tested:

read

This command calls the device driver ReadFile entry point and attempts to read from the tape device special file opened previously. The amount of data to be read will depend of the current settings of the *block_size* and *block_count* variables and on the number of records specified.

This command transfers *n* records. A record will have a length of *block_size*block_count* for fixed block mode and *block_count* for variable mode. For example, if *block_size=1024*, *block_count=2* and *records=3*, then three 2048 transfers will be done. If the operation succeeds, the RC will show the total number of bytes transferred.

SYNTAX: read
read records = n

for example, read records = 2

FUNCTION Tested:

read_block_id

This command calls the device driver GetTapePosition entry point with an operation of TAPE_LOGICAL_POSITION. This attempts to read the current block ID value for the cartridge in the tape device special file opened previously.

SYNTAX: read_block_id

for example, read_block_id

FUNCTION Tested:

read_dump

This command calls the device driver DeviceIoControl entry point with an operation of IOCTL_SCSI_PASS_THROUGH and a CDB[0] of SCSIOP_READ_DATA_BUFF. This reads the dump data and writes it to a file specified by *dump_name* or to a *dump0000.dmp* default file. *Dump_name* must be

eight characters or less. *Ntutil* will add the .dmp extension automatically. The dump is written into the directory where *ntutil* was started.

SYNTAX: read_dump
read_dump dump_name=string device=devnum
where string is eight characters or less and devnum=0 (for tape) or 1 (for changer)

For example, read_dump dump_name = fsc0000 device = 0.

FUNCTION Tested:

read_write (test)

This command performs the following test on the cartridge in the tape device special file opened previously:

1. Rewind the tape.
2. Set Media Parameters to fixed block 32 KB.
3. Write a 32 KB random data block on tape.
4. Rewind the tape.
5. Read the block and compare the data to the data written in step 3.

SYNTAX: read_write

for example, read_write

FUNCTION Tested:

restore_data

This command calls the device driver ReadFile entry point. This attempts to perform 64 KB variable block read operations on the tape device special file until either a filemark or an end of data is detected. All the data read, regardless of error, is saved in the file specified. If there is no data on the tape, the command will fail, and no data will be saved. The RC should be ignored in most cases, because this operation always ends with an error (either filemark detected or end of data detected).

SYNTAX: restore_data
restore_data file_name=string

for example, read_data file_name=fsc0000

FUNCTION Tested:

return_error

This command calls GetLastError. If the command is successful, the information returned will be displayed. There is no RC for this function.

SYNTAX: return_error_state

for example, return_error_state

FUNCTION Tested:

return_lib_inventory_all

This command calls the device driver DeviceIoControl entry point with an opcode of LIBRARY_INVENTORY and attempts to return information about all the hardware components in a library. If this command is successful, the information returned will be displayed. For Windows NT, see the appropriate hardware manuals for a description of the information returned by a read element status command.

In Windows 200x, RSM must be stopped to issue this command. A remapped inventory is returned, where one element is returned per line.

Windows Utility Program (ntutil)

Each line has the following format:

Type: *n*, Addr: hex-value, “Empty” or “Full”, access-type on a new line

- *n* is 2, 3, or 4, which indicate SE (ChangerSlot), IEE (ChangerIEPort), or DTE (ChangerDrive), respectively.
- hex value is a hexadecimal value that indicates the mapped address of the item. This mapped address is used for medium operations on Windows 200x, which ID differently from using the actual hardware element addresses on Windows NT.
- *Empty* or *Full* indicates whether the element is occupied. Note that if a cartridge is in the drive, *Empty* is returned.
- access-type will be set to *Access* if the element is accessible, blank if not.

If tracing is enabled, the flags field for the element will be displayed after the element address. Potential values for this field are documented in the ELEMENT_STATU_abc constants in *ntddchgr.h*.

SYNTAX: return_lib_inventory_all

for example, return_lib_inventory_all

FUNCTION Tested:

reverse_filemark

This command calls the device driver SetTapePosition entry point with an operation of TAPE_SPACE_FILEMARKS and a count of *n* and attempts to backward space *n* files on the cartridge in the tape device special file opened previously.

SYNTAX: reverse_filemark *n*

for example, reverse_filemark 2

FUNCTION Tested:

reverse_record

This command calls the device driver SetTapePosition entry point with an operation of TAPE_SPACE_RELATIVE_BLOCKS and a count of *n* and attempts to backward space *n* records on the cartridge in the tape device special file opened previously.

SYNTAX: reverse_record *n*

for example, reverse_record 2

FUNCTION Tested:

rewind

This command calls the device driver SetTapePosition entry point with an operation of TAPE_REWIND and attempts to rewind the tape device special file opened previously. Status is presented when the rewind is complete.

SYNTAX: rewind

for example, rewind

FUNCTION Tested:

set_device_parameters (compression)

This command calls the device driver SetTapeParameters entry point with an operation of SET_TAPE_DRIVE_INFORMATION. The compression value is a required parameter with the command.

compression - device compression On or Off (0 for Off, any other value On).

SYNTAX: set_device_parameters compression = n

for example, set_device_parameters compression = 0

FUNCTION Tested:

set_media_parameters (block size)

This command calls the device driver SetTapeParameters entry point with an operation of SET_TAPE_MEDIA_INFORMATION. The *block_size* value is a required parameter with the command.

block_size - device block size setting (0 for variable).

SYNTAX: set_media_parameters block_size = n

for example, set_media_parameters block_size = 0

FUNCTION Tested:

space_eod

This command calls the device driver SetTapePosition entry point with an operation of TAPE_SPACE_END_OF_DATA and attempts to space to the end of data on the cartridge in the tape device special file opened previously.

SYNTAX: space_eod

for example, space_eod

FUNCTION Tested:

test_unit_ready

This command calls the device driver DeviceIoControl entry point with an operation of IOCTL_SCSI_PASS_THROUGH and a CDB[0] of SCSIOP_TEST_UNIT_READY. If the operation is not successful, the sense data will be displayed. Specify 0 to send the command to the drive. Specify 1 to send the command to the changer.

SYNTAX: test_unit_ready n

for example, test_unit_ready 1

FUNCTION Tested:

unload

This command calls the device driver PrepareTape entry point with an operation of TAPE_UNLOAD. This attempts to rewind and unload the tape device special file opened previously.

SYNTAX: unload

for example, unload

FUNCTION Tested:

update_code

This command is used to update the microcode and calls the device driver DeviceIoControl entry point with an operation of IOCTL_SCSI_PASS_THROUGH and a CDB[0] of SCSIOP_WRITE_DATA_BUFF. The filename portion of the image

Windows Utility Program (ntutil)

name must be specified, including the extension. You must indicate the full pathname to the file unless the file resides in the current working directory (for example, the directory that was open at the time *ntutil* was started). The image filename extension must be **.fmr**.

SYNTAX: `update_code image_name = string`

for example, `update_code image_name = d0i9_430`

FUNCTION Tested:

write

This command calls the device driver WriteFile entry point and attempts to write to the tape device special file opened previously. The amount of data to be written will depend of the current settings of the *block_size* and *block_count* variables and on the number of records specified.

This command transfers **n** records. A record will have a length of `block_size * block_count` for fixed block mode, and `block_count` for variable mode. For example, if `block_size=1024`, `block_count=2` and `records=3`, then three 2048 transfers will be done. If the operation succeeds, the RC will show the total number of bytes transferred.

If the records attribute is not specified, the default is 1.

Each record written will have random bytes preceded by an integer identifier. If the data attribute is not specified, the identifier will be a unique sequential counter. This can be overridden by specifying **data=**. You can look for this value on a read command by specifying, for example, `result data=888`. If data is specified, the record count **must** be specified. Data can be a decimal value from 0 to 4294967295.

SYNTAX: `write`
`write records = n`
`write records = n data = m`

for example, `write records = 3 data = 888`

FUNCTION Tested:

write_filemark

This command calls the device driver WriteTapemark entry point with an operation of TAPE_FILEMARKS and a count of **n**. This attempts to write **n** filemarks on the tape media.

SYNTAX: `write_filemark n`

for example, `write_filemark 2` (write 2 file marks)

FUNCTION Tested:

Supported Result Text Fields

The result text on a command statement tests the completion status for an expected result. Further execution of a test case is affected by the current settings of the *return_error_when_fail* and *exit_on_unexpected_result* variables. These can be set using a *set* statement.

If no result text is specified, the command will always be treated as successful.

In the case of rc, except as noted in the command section, a failure will return -1, else the operation succeeded.

The syntax for result text is:

```
< rc < n> <err < n> <cc < n>
>
<=
>=
==
!=
>
>=
==
!=
>
>=
==
!=
```

where:

rc = return code from the last command.
 err = Operating System GetLastError() value for the last command.
 data = expected record identifier for a read command.

For example, consider this statement in the input file:

```
command write 1 result rc == -1 err == WRITE_PROTECT
```

This means we will try to write one record to tape and we expect this to fail with return code -1, err WRITE_PROTECT.

Note: Symbolic values or integers can be used for *err*. For a list of recognized symbolic values, see Figure 45 on page 273.

set

This statement allows setting of variables that affect how tests are executed. The syntax for a set statement is:

```
set variable = value
```

for example, set return_error_when_fail = 1

The following variables can be set:

return_error_when_fail

Can be set to 1 (TRUE) or 0 (FALSE). A setting of TRUE means *ntutil* will end by returning an error at the end of the test if any command does not produce a correct expected result as specified in result text.

exit_on_unexpected_result

This can be set to 1 (TRUE) or 0 (FALSE). If set to TRUE, the first occurrence of an unexpected result will terminate the test.

block_id

This can be set to a four byte hex value (for example, 0001aa03). This value is used on a *locate_block_id* command.

block_size

This is the block size to be used for *reading*, *writing*, or *set_mode*. It can be set to 0 to indicate variable block mode.

type

Text following the word **type** is typed on the terminal. This can be used to show the progress of the test or to prompt for a subsequent pause statement.

```
type string
```

for example, type tell operator it's lunch time

Windows Utility Program (ntutil)

pause

The test stops until a character is typed at the terminal. This allows for tests that require manual intervention.

delay

The test stops for **n** seconds. This allows for tests that require delay for mechanical completion before continuing. The syntax for the delay statement is:

delay value

for example, delay 1

system

Text following the word system is passed to the system for execution. The syntax for the system statement is:

system operation

for example, system dir

symbols

This statement prints the symbol list that can be used in result text fields.

exit

This statement causes immediate termination of the batch test. No further lines will be read from the input file.

Symbolic Values

Figure 45 on page 273 shows the values (symbolic error or error number) that can be used for *err*.

Symbolic Error	NT/Win2000 Error Number
ERROR_SUCCESS	0
ERROR_INVALID_FUNCTION	1
ERROR_FILE_NOT_FOUND	2
ERROR_PATH_NOT_FOUND	3
ERROR_TOO_MANY_OPEN_FILES	4
ERROR_ACCESS_DENIED	5
ERROR_INVALID_HANDLE	6
ERROR_NOT_ENOUGH_MEMORY	8
ERROR_BAD_FORMAT	9
ERROR_INVALID_BLOCK	10
ERROR_BAD_ENVIRONMENT	11
ERROR_INVALID_ACCESS	12
ERROR_INVALID_DATA	13
ERROR_OUTOFMEMORY	14
ERROR_INVALID_DRIVE	15
ERROR_WRITE_PROTECT	19
ERROR_BAD_UNIT	20
ERROR_NOT_READY	21
ERROR_BAD_COMMAND	22
ERROR_CRC	23
ERROR_HANDLE_EOF	38
ERROR_NOT_SUPPORTED	50
ERROR_DEV_NOT_EXIST	55
ERROR_ALREADY_ASSIGNED	85
ERROR_INVALID_PARAMETER	87
ERROR_OPEN_FAILED	110
ERROR_INSUFFICIENT_BUFFER	122
ERROR_INVALID_NAME	123
ERROR_BUSY_DRIVE	142
DD_NO_SENSE	200
DD_DEVICE_DRIVER_FAILURE	201
DD_EEPROM_FAILURE	202

Figure 45. Symbolic Values (Part 1 of 3)

Windows Utility Program (ntutil)

Symbolic Error	NT/Win2000 Error Number
DD_MANUAL_INTERVENTION	203
DD_RECOVERED_ERROR	204
DD_SCSI_ADAPTER_ERROR	205
DD_SCSI_ERROR	206
DD_SCSI_BUSY	211
DD_ILLEGAL_REQUEST	207
DD_COMMAND_ABORTED	208
DD_HARDWARE_MICROCODE	209
DD_UNIT_ATTENTION	210
ERROR_MORE_DATA	234
DD_CARTRIDGE_ENTRY_FAILURE	300
DD_CARTRIDGE_LOAD_FAILURE	301
DD_CARTRIDGE_IN_FAILED_DRIVE	302
DD_CAROUSEL_NOT_LOADED	303
DD_CHANGER_FAILURE	304
DD_DRIVE_FAILURE	305
DD_DRIVE_OR_MEDIA_FAILURE	306
DD_ENTRY_EXIT_FAILURE	307
DD_ENTRY_EXIT_NOT_PRESENT	308
DD_LIBRARY_AUDIT	309
DD_LIBRARY_FULL	310
DD_MEDIA_EXPORT	311
DD_SLOT_FAILURE	312
DD_SLOT_OR_MEDIA_FAILURE	313
DD_SOURCE_EMPTY	314
DD_DESTINATION_FULL	315
DD_CLENER_INST	316
DD_MEDIA_NOT_EJECTED	317
DD_IOPORT_NOT_CONFIG	318
DD_FIRST_DEST_EMPTY	319
DD_END_PHYSICAL_MEDIA	400
DD_MEDIA_BLANK	401
DD_MEDIA_CORRUPTED	402
DD_MEDIA_FAILURE	403
DD_MEDIA_INCOMPATIBILITY	404
DD_SECTOR_RELOCATION	405
DD_SECTOR_OUT_OF_RANGE	406
DD_WRITE_PROTECT	407
DD_CLEAN_MEDIA	408
DD_MEDIA_FAULT	409
DD_CLEANING_COMPLETE	410
DD_LOGICAL_END_OF_MEDIA	411
DD_MEDIA_NOT_PRESENT	412
DD_BEGINNING_OF_MEDIA	413

Figure 45. Symbolic Values (Part 2 of 3)

Symbolic Error	NT/Win2000 Error Number
DD_ERASE_FAILURE	414
DD_WRITE_TO_WRITTEN_WORM	415
DD_WRONG_LENGTH_BLOCK	416
ERROR_IO_INCOMPLETE	996
ERROR_IO_PENDING	997
ERROR_NOACCESS	998
ERROR_CANTOPEN	1011
ERROR_CANTREAD	1012
ERROR_CANTWRITE	1013
ERROR_END_OF_MEDIA	1100
ERROR_FILEMARK_DETECTED	1101
ERROR_BEGINNING_OF_MEDIA	1102
ERROR_SETMARK_DETECTED	1103
ERROR_NO_DATA_DETECTED	1104
ERROR_PARTITION_FAILURE	1105
ERROR_INVALID_BLOCK_LENGTH	1106
ERROR_DEVICE_NOT_PARTITIONED	1107
ERROR_UNABLE_TO_LOCK_MEDIA	1108
ERROR_UNABLE_TO_UNLOAD_MEDIA	1109
ERROR_MEDIA_CHANGED	1110
ERROR_BUS_RESET	1111
ERROR_NO_MEDIA_IN_DRIVE	1112
ERROR_IO_DEVICE	1117
ERROR_TOO_MANY_LINKS	1142

Figure 45. Symbolic Values (Part 3 of 3)

Device Driver Diagnosis Information (for Microsoft Windows NT)

There is a debug version of the device driver that can be used if you encounter difficulties with the device driver. The debug version of the driver issues *DbgPrint* messages at various places during device driver execution. To capture these messages, you will need to start a debugger or use a tool like *DebugView*, available from:

<http://www.sysinternals.com>

To install and use the debug version of the device driver, perform the following steps:

1. Quiesce all activity on Ultrium devices.
2. Exit all applications using Ultrium devices.
3. Stop the Ultrium device driver (*IBMtape.sys*). See “Manual Starting and Stopping Procedures” on page 236 for details on stopping the driver.
4. Find the ... \checked directory for the device driver level that you are running. If you chose the default installation location when you installed the drivers, the debug drivers will be found at *c:\Program files\IBM Corporation\IBM Ultrium Device Drivers\n.n.n.n\checked*, where n.n.n.n are integers that indicate the driver level. To determine the driver level, find *c:\winnt\system32\drivers\ibmtape.sys* using Windows Explorer, right-click the file, select **Properties**, then select the **Version** tab. If you do not find a ... \checked directory for your driver, you must install the **Device Driver Depot** component of the Ultrium Device Driver Installation Package.
5. Go to the root prompt, for example, c:

Windows Utility Program (ntutil)

6. Change to the appropriate directory by typing `cd \winnt\system\drivers`
7. Copy `IBMtape.sys ibmtape.orig`
8. Copy `IBMtape.sys` from the appropriate ... \checked directory to `c:\winnt\system32\drivers\ibmtape.sys`. For example, if you installed the device drivers at the default installation location and your driver level was 1.1.6.9, you would issue the following command: `copy c:\Program files\IBM Corporation\IBM Ultrium Device Drivers\1.1.6.9\checked\IBMtape.sys c:\winnt\system32\drivers\ibmtape.sys`.
9. Start the debugger or tool that captures the `DbgPrint` messages.
10. Start the Ultrium device driver. See “Manual Starting and Stopping Procedures” on page 236 for details on starting the driver.

Registry variable

`HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Ultrium/MsgLevel` determines how many messages are issued by the debug version of the driver. When the debug version of the driver is started for the first time, it defines the registry variable. This variable is ignored by the non-debug version of the driver.

The customer can set the value of this variable, using `regedt32`, to any value from `REG_DWORD 0x0` to `REG_DWORD 0x5`, inclusive. The greater the value, the more messages will be issued.

To restore the non-debug version of the driver, perform the following steps:

1. Quiesce all activity on Ultrium devices.
2. Exit all applications using Ultrium devices.
3. Stop the Ultrium device driver (`IBMtape.sys`). See “Manual Starting and Stopping Procedures” on page 236 for details on stopping the driver.
4. Go to the root prompt, for example, `c:`
5. Change to the appropriate directory by entering `cd \winnt\system\drivers`
6. Copy `ibmtape.orig IBMtape.sys`
7. Start the Ultrium device driver. See “Manual Starting and Stopping Procedures” on page 236 for details on starting the driver.

The link maps for the driver and the debug version of the driver are stored on the installation diskette in the maps directory. The `ibmtape.fre` file is the map for the regular version of the driver. The `ibmtape.chk` file is the map for the debug version of the driver.

Device Driver Diagnosis Information (for Microsoft Windows 200x)

There is a debug version of the device driver that can be used if you encounter problems. The debug version of the driver issues `DbgPrint` messages at various places during device driver execution. To capture these messages, you must start a debugger or use a tool like `DebugView`, available from:

<http://www.sysinternals.com>

Using the Debug Version

To install and use the debug version of the device driver, perform the following steps after the driver has initially been installed:

1. Quiesce all activity on Ultrium devices.
2. Exit all applications that are using Ultrium devices.

3. Locate the *\checked* folder for the device driver level that you are running. This folder is in the highest level folder of the zip archive and it contains checked versions of *ibmtpxxx.sys* and *ibmcgxxx.sys* (where *xxx* = 2k for Windows 2000 and Windows Server 2003, 32-bit or 2k3 for Windows Server 2003, 64-bit). Copy the checked version of *ibmtpxxx.sys* or *ibmcgxxx.sys* to *\winnt\system32\drivers*, overwriting the version of the file already there.
4. Reboot the system.
5. Start the debugger to capture the DbgPrint messages.
6. Issue the commands to the driver. You should see debug statements printed to the debugger window.

Restoring the Non-Debug Version

To restore the non-debug version of the driver, perform the following steps:

1. Quiesce all activity on Ultrium devices.
2. Exit all applications that are using Ultrium devices.
3. In the highest level directory of the zip archive, you will find non-debug versions of *ibmtpxxx.sys* and *ibmcgxxx.sys* (where *xxx* = 2k for Windows 2000 and Windows Server 2003, 32-bit or 2k3 for Windows Server 2003, 64-bit). Copy the non-debug version of *ibmtpxxx.sys* or *ibmcgxxx.sys* to *\winnt\system32\drivers*, overwriting the version of the file already there.
4. Reboot the system. When the driver has started and commands are issued to it, the driver will no longer produce debug output.

Windows Utility Program (ntutil)

Part 8. 3494 Enterprise Tape Library Emulation

Chapter 45. 3494 Library Emulation Support for SCSI Medium Changers

Overview

The 3494 Library Emulation support provides the ability to use applications that were written for the 3494 API on host attached SCSI Medium Changer Libraries. The emulation supports any type of SCSI Medium Changer library and drive attached such as a 3584 Ultra Scalable Library with either 3592 or LTO Ultrium Drives.

Existing customer scripts using the mtlib program and customer or ISV applications will require no or minor changes to use, depending on the 3494 functions being used. Customer and ISV applications will only need to be recompiled with the new libibm.o object module in order to use the emulation support. The mtlib 3494 user command line interface program that has the same syntax and output on every O/S platform will provide a common utility for SCSI Medium Changer Libraries also. Current O/S device driver utilities vary by syntax/output, require a knowledge of SCSI Medium Changers to use, and require multiple commands to perform a single mtlib command.

3494 Emulation Design

The 3494 Emulation support adds a libsmc extension to the current libibm.o object module that applications compile with for existing 3494 support. The libsmc extension contains the 3494 API Emulation support, which issues SCSI Medium changer commands to the library using the O/S SCSI Medium Changer device driver.

Figure 46 and Figure 47 on page 282 show the 3494/SMC data flow between existing applications and components and the new libsmc extension.

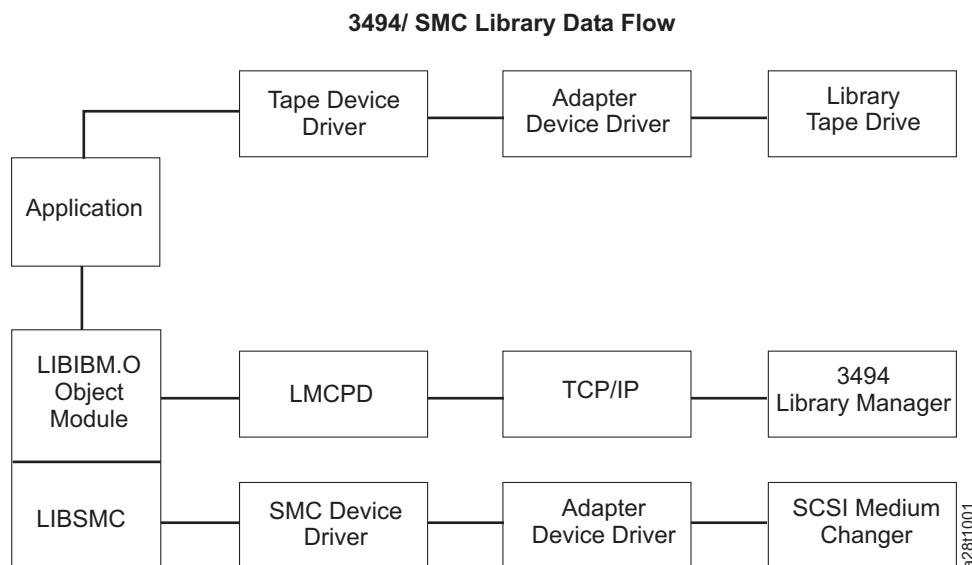


Figure 46. 3494/SMC Library Data Flow

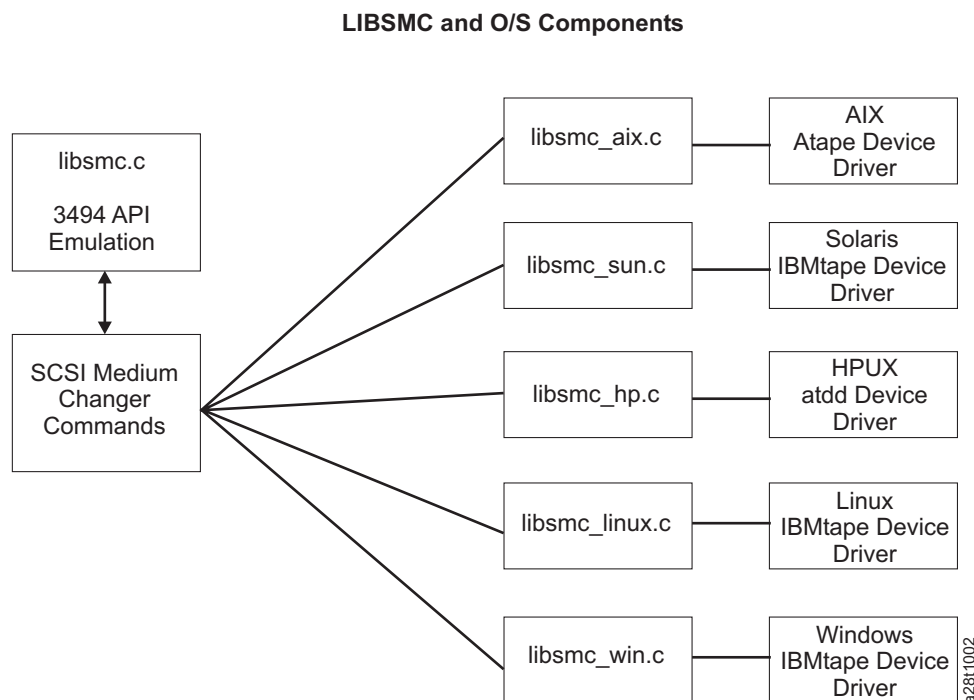


Figure 47. LIBSMC and OS Components

Using the 3494 API Emulation and MTLIB Program

Refer to the *IBM TotalStorage Tape Device Drivers Installation and User's Guide* and the *IBM TotalStorage Tape Device Drivers Programmer's Reference* for the complete description of 3494 library support and the `mtlib` program and syntax.

SMC Library Names

The library name on the `open_ibmatl` ("libname") or `mtlib -l libname` parameter determines whether a 3494 library or an SMC library is being used. If the library name is a logical O/S SMC library name then the `libsmc` support will be used. Otherwise, the `lmcpd` support will be used.

For example, SMC libraries on AIX are named `smc0`, `smc1`, and so on. To issue an `mtlib` query library command on AIX to a 3584 library named `smc0`:

```
mtlib -l smc0 -qL
```

To issue an `mtlib` command on Linux to a 3584 library named `IBMchanger0`:

```
mtlib -l IBMchanger0 -qL
```

Volume Categories

The 3494 library supports the grouping of one or more volumes into categories that can be assigned by an application and operations that can specify a category only rather than a specific volume. SCSI Medium Changers do not provide a similar function. All volumes in an SMC library are category 0000, the NULL category.

Because volume categories are not supported, the following API and **mtlib** command options cannot be used. Applications and **mtlib** scripts that use these functions might require changes.

- Reserve category **mtlib -r** option
- Release category **mtlib -R** option
- Change a volume category **mtlib -C** option
- Set category attribute **mtlib -S** option
- Assign a category **mtlib -k** option
- Mount from a category **mtlib -m -s** option

Asynchronous Library Operations

The 3494 library performs all mount, demount, audit, and eject operations asynchronously and initially returns a request id to the host for the operation. The request id can then be queried at any time to determine if the operation is still pending, completed, or is unknown. SMC libraries perform these operations synchronously and do not support a request id.

An application can select to wait for an asynchronous operation to complete by setting the API `wait_flg` to one or using **mtlib** without the `-n` option. The operation will be synchronous on both a 3494 and SMC library and no request id is returned to the application. These applications and **mtlib** scripts will not require any changes.

If an application selects to not wait for an asynchronous operation to complete by setting the API `wait_flg` to zero or using **mtlib** with the `-n` option, the operation on an SMC library will still be synchronous and request id 0 is returned to the application.

Most applications and scripts do not use the return request id to determine when the operation completes and use the no wait option for performance only. For example, using the no wait option on a demount operation so the application can continue while the library is demounting the volume. These applications and **mtlib** scripts will not require any changes but will take the same amount of time as using the wait option.

Applications that use the Query Message ID or **mtlib -i** option to determine when the return request id operation completes may require changes, such as polling for a mount issued with the no wait option since any request id will always be unknown on an SMC library because the operation has already completed.

Part 9. Appendixes

Appendix A. Accessing Documentation and Software Online

IBM maintains the latest levels of Ultrium tape drive and library device drivers and documentation on the Internet. Obtain them by accessing the following URL:

<ftp://ftp.software.ibm.com/storage/devdvr/tapedrivers.html>

This web page has links to the device driver download web pages.

Information concerning supported fibre channel host bus adapters (HBAs), and associated HBA device drivers, firmware and BIOS levels can be obtained from the following URL:

<http://knowledge.storage.ibm.com/HBA/HBASearchTool>

You can also browse the FTP site and download drivers using one of the following URLs:

<ftp://ftp.software.ibm.com/storage/devdvr>

<ftp://207.25.253.26/storage/devdvr>

Or, you can access this information using anonymous FTP, as follows::

FTP site: [ftp.software.ibm.com](ftp://ftp.software.ibm.com)

IP Address: 207.25.253.26

User ID: anonymous

Password: (Use your current e-mail address.)

Directory: /storage/devdvr

IBM provides Postscript- and PDF-formatted versions of its documentation in the */storage/devdvr/Doc* directory. Postscript documents are gzip-ed.

The *IBM_ultrium_tape_IUG.pdf* and *IBM_ultrium_tape_IUG.ps.gz* files contain the current version of the *IBM Ultrium Device Drivers: Installation and User's Guide*.

The *IBM_ultrium_tape_PROGREF.pdf* and *IBM_ultrium_tape_PROGREF.ps.gz* files contain the current version of the *IBM Ultrium Device Drivers: Programming Reference*.

Device and Library Drivers (except Tru64, which just provides a utility program) for each supported platform can be found beneath */storage/devdvr/* in the following directories:

AIX/

HPUX/

Linux/

Solaris/

Tru64/

Windows/

There are numeric sequence numbers in each level of device and library driver, such as *AIX/Atape.4.4.0.0.bin*. When a new level of a driver is released, a higher numeric sequence is assigned.

The following table documents each driver and/or utility by name and description:

Note: The *n.n.n.n.* strings are replaced with digits on the FTP site to reflect the version of each driver. *dist* indicates a Linux distribution.

Accessing Documentation and Software Online

AIX/Atape.n.n.n.n.bin	AIX Device Driver (Atape)
HPUX/atdd.n.n.n.n.bin	HP-UX Device Driver (ATDD)
Linux/dist/IBMtape.n.n.n.i386.rpm.bin	Linux Device Driver (IBMtape)
Linux/dist/IBMtapeutil.n.n.n.i386.tar.bin	Linux Tape Utility (IBMtapeutil)
Solaris/IBMtape.n.n.n.n.bin	Solaris Device Driver (IBMtape)
Tru64/RasUtil.n.n.n.kit.tar	Tru64 RAS Utility Program
Windows/WinNT/IBMUltrium.WinNT.exe	Windows NT device Driver (IBMtape)
Windows/Win2000/IBMUltrium.Win2K_nnnn.zip	Windows 2000 Device Driver (IBMtape)
Windows/Win2003/IBMUltrium.W2K3_32_nnnn.zip	Windows 2003 Device Driver, 32-bit (IBMtape)
Windows/Win2003/IBMUltrium.W2K3_64_nnnn.zip	Windows 2003 Device Driver, 64-bit (IBMtape)

Appendix B. Verifying Proper Attachment of Your Devices

Before you start to use your devices for production work with your applications, or if you encounter difficulties with your devices, you might want to verify that the hardware, connections, and device drivers are working together properly. Before you can do this, you must do the following:

1. Install your hardware as indicated in the appropriate hardware manuals.
2. Power on your hardware and verify that the hardware is functioning properly by executing commands from the front panel.
3. Attach your hardware to the host system as indicated in the appropriate hardware manuals and as indicated in the appropriate chapters from this manual.
4. Start your operating system as indicated in the appropriate chapters from this manual.
5. Log on the operating system as Administrator.
6. If your device is using device drivers other than the ones documented in this manual, disable the other device drivers and install or enable the drivers documented in this manual.
7. Follow the subsequent instructions for your host system.

AIX System

Tape Device Attachment Test

The following procedure tests the attachment of a tape device to an AIX system. The procedure assumes that your device is attached at `/dev/rmt0` and that there is no cartridge in the drive.

When the **Hit <Enter> to Continue . . . ?** message is displayed or when you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Open an AIX window.
2. If you want to perform a more complete test and your device has an autoloader, use the instructions in “Medium Changer Device Attachment Test” on page 290 to mount a writable scratch cartridge in the drive. This is accomplished by following the steps in the procedure, except that in steps 11 and 12, select the element ID of a drive rather than the unoccupied slot ID.
If you want to perform a more complete test and your device does not have an autoloader, mount a writable scratch cartridge manually into the driver.
3. Enter `tapeutil`. A menu will be displayed.
4. Enter **1** (Open a Device).
5. Enter `/dev/rmt0` when prompted for the device name.
6. Enter **1** (Read/Write).
7. Enter **3** (Device Info).
8. Enter **5** (Inquiry). Specify **0** when prompted for an inquiry page. This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.

Verifying Proper Attachment of Your Devices

9. Enter **6** (Test Unit Ready) until no error occurs.
10. Enter **22** (Rewind).
11. Enter **30** (Read and Write Tests).
12. Enter **1** (Read/Write). Press the **Enter** key three times to accept the defaults and run the test.
13. Enter **31** (Unload Tape).
14. Enter **2** (Close a Device).
15. Enter **Q** (Quit Program).

Medium Changer Device Attachment Test

The following procedure tests the attachment of a medium changer device to an AIX system. The procedure assumes that your device is attached at `/dev/rmt0`. You also need a cartridge in at least one of the slots.

When the **Hit <Enter> to Continue . . .?** message is displayed or when you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Open an AIX window.
2. Enter `tapeutil`. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter `/dev/rmt0` when prompted for the device name.
5. Enter **1** (Read/Write).
6. Enter **3** (Device Info).
7. Enter **5** (Inquiry). Specify **0** when prompted for an inquiry page. This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.
8. Enter **6** (Test Unit Ready) until no error occurs.
9. Enter **18** (Initialize Element Status).
10. Enter **14** (Element Inventory).
11. From the output of the previous step, select a writable, scratch cartridge and determine its element ID. Also, select the element ID of an unoccupied slot.
12. Enter **16** (Move Medium), then supply the address of the cartridge, followed by the address of the unoccupied slot. Verify that the cartridge moved.
13. Enter **14** (Element Inventory). Verify that the inventory was updated properly.
14. Enter **2** (Close a Device).
15. Enter **Q** (Quit Program).

Compaq Tru64 System

Tape Device Attachment Test

The following procedure tests the attachment of a tape device to a Compaq Tru64 system. The procedure assumes that your device is attached at `/dev/ntape/tape0`.

When you are asked to enter information, press the **Enter** key after following the instructions. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Logon to the system as `root`, or use the `su` command to gain superuser privileges.

2. Bring up a shell prompt.
3. Enter ***/sbin/scu -f /dev/ntape/tape0*** (Open the tape device).
4. Enter ***tur*** (Test Unit Ready).
5. Enter ***show inquiry*** (Inquiry data).
6. Enter ***quit*** to quit *scu* Medium Changer Device Attachment Test

Medium Changer Device Attachment Test

The following procedure tests the attachment of a medium changer device to a Compaq Tru64 system. The procedure assumes that your device is attached at */dev/changer/mc0*. You also need a cartridge in at least one of the slots. When you are asked to enter information, press the **Enter** key after following the instructions. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Logon to the system as *root*, or use the *su* command to gain superuser privileges.
2. Bring up a shell prompt.
3. Enter ***/sbin/scu -f /dev/changer/mc0*** (Open the medium changer device).
4. Enter ***show inquiry*** (Inquiry Data).

This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.
5. Enter ***tur*** (Test Unit Ready) until no error occurs.
6. Enter ***show element*** (Element Inventory).
7. From the output in the previous step, select a writable, scratch cartridge and determine its element address. Also select the element address of an unoccupied slot.
8. Enter ***move medium source Source_address destination Destination_address*** to move the cartridge from the source element address to the destination element address. Verify that the cartridge moved.
9. Enter ***show element*** (Element Inventory) to verify that the inventory was updated properly.
10. Enter ***quit*** (Quit *scu*).

HP-UX System

Tape Device Attachment Test

The following procedure tests the attachment of a tape device to an HP-UX system. The procedure assumes that your device is attached at */dev/rmt/0m*.

When you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Bring up a shell prompt.
2. Enter ***tapeutil***. A menu will be displayed.
3. Enter ***1*** (Open a Device).
4. Enter ***/dev/rmt/0m*** when prompted for the device name.
5. Enter ***1*** (Read/Write).
6. Enter ***D*** (Device Info).
7. Enter ***10*** (INQUIRY option).
8. Enter ***Q*** to quit *tapeutil*.

Autochanger Device Attachment Test

The following procedure tests the attachment of an autochanger device to an HP-UX system. The procedure assumes that your device is attached at `/dev/rmt/1chnng`. You also need a cartridge in at least one of the slots.

When you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Bring up a shell prompt.
2. Enter `tapeutil`. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter `/dev/rmt/1chnng` when prompted for the device name.
5. Enter **1** (Read/Write).
6. Enter **D** (Device Info).
7. Enter **10** (INQUIRY option). This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.
8. Enter **9** (TEST UNIT READY option) until no error occurs.
9. Enter **19** (ELEMENT INVENTORY option).
10. From the output in the previous step, select a writable, scratch cartridge and determine its element ID. Also, select the element ID of an unoccupied slot.
11. Enter **16** (MOVE MEDIUM option), then supply the address of the cartridge and the address of the unoccupied slot. Verify that the cartridge moved.
12. Enter **19** (ELEMENT INVENTORY option). Verify that the inventory was updated properly.
13. Enter **2** (Close a Device).
14. Enter **Q** (Quit Program).

Linux System

Issue the following command to verify if your host system has recognized all the attached IBM tape and medium changer devices after you installed the IBMtape device driver:

```
cat /proc/scsi/scsi
```

You can follow the instructions below to test the attachment.

Tape Device Attachment Test

The following procedure tests the attachment of a tape device to a Linux system. The procedure assumes that your device is attached at `/dev/IBMtape0` and that there is no cartridge in the drive.

When the **Hit<Enter>to Continue...** message is displayed or when you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Open a Linux window.
2. If you want to perform a more complete test, mount a writable scratch cartridge manually into the driver, or, if your device has an autoloader attached, follow the "Medium Changer Device Attachment Test" in the next section to mount a writable scratch cartridge.

3. Enter *IBMtapeutil*. A menu will be displayed.
4. Enter **1** (Open a Device).
5. Enter */dev/IBMtape0* when prompted for the device name.
6. Enter **1** (Read/Write).
7. Enter **3** (Inquiry). Specify **0** when prompted for an inquiry page. This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test as follows:
 8. Enter **4** (Test Unit Ready) until no error occurs.
 9. Enter **20** (Rewind).
 10. Enter **28** (Read and Write Tests).
 11. Enter **1** (Read/Write). Press the **Enter** key three times to accept the defaults and run the test.
 12. Enter **38** (Unload Tape).
 13. Enter **2** (Close a Device).
 14. Enter **Q** (Quit Program).

Medium Changer Device Attachment Test

The following procedure tests the attachment of a medium changer device to a Linux system. The procedure assumes that your device is attached at */dev/IBMchanger0*. You also need a cartridge in at least one of the slots.

When the **Hit <Enter> to Continue ...** message is displayed or when you are told to enter information, press the **Enter** key after following the instructions. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Open a Linux window.
2. Enter *IBMtapeutil*. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter */dev/IBMchanger0* when prompted for the device name.
5. Enter **3** (Inquiry). Specify **0** when prompted for an inquiry page. This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test as follows:
 6. Enter **4** (Test Unit Ready).
 7. Enter **60** (Element Information).
 8. Enter **62** (Element Inventory).
 9. From the output of the previous step, select a writable, scratch cartridge and determine its element ID. Also, select the element ID of an unoccupied slot.
 10. Enter **64** (Move Medium), then supply the address of the cartridge, followed by the address of the unoccupied slot. Verify that the cartridge moved.
 11. Enter **62** (Element Inventory). Verify that the inventory was updated correctly.
 12. Enter **2** (Close a Device).
 13. Enter **Q** (Quit Program).

Solaris System

Tape Device Attachment Test

The following procedure tests the attachment of a tape device to a Solaris system. The procedure assumes that your device is attached at `/dev/rmt/0st`. When you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Bring up a shell prompt.
2. Enter `tapeutil`. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter `/dev/rmt/0st` when prompted for the device name.
5. Enter **1** (Read/Write).
6. Enter **D** (Device Info).
7. Enter **10** (Inquiry).
8. Enter **Q** to quit `tapeutil`.

Autochanger Device Attachment Test

The following procedure tests the attachment of an autochanger device to a Sun Solaris system. The procedure assumes that your device is attached at `/dev/rmt/0st`. You also need a cartridge in at least one of the slots.

When you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Bring up a shell prompt.
2. Enter `tapeutil`. A menu will be displayed.
3. Enter **1** (Open a device).
4. Enter `/dev/rmt/0st` when prompted for the device name.
5. Enter **1** (Read/Write).
6. Enter **D** (Device Info).
7. Enter **5** (Inquiry). This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.
8. Enter **6** (Test Unit Ready) until no error occurs.
9. Enter **17** (Element Inventory).
10. From the output in the previous step, select a writable, scratch cartridge and determine its element ID. Also, select the element ID of an unoccupied slot.
11. Enter **14** (Move Medium), then supply the address of the cartridge and the address of the unoccupied slot. Verify that the cartridge moved.
12. Enter **17** (Element Inventory).
13. Verify that the inventory was updated properly.
14. Enter **2**(Close a Device).
15. Enter **Q** (Quit Program).

Microsoft Windows System

Tape Device Attachment Test

The following procedure tests the attachment of a tape device to a Microsoft Windows system. The procedure assumes that your device is attached at *tape0* and that there is no cartridge in the drive.

When the **Return to continue:** message is displayed or when you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Open a Windows command shell window.
2. If you want to perform a more complete test and your device has an autoloader, use the instructions in “Autochanger Device Attachment Test - Windows NT only” to mount a writable scratch cartridge in the drive. This is accomplished by following the steps in the procedure, except that in steps 11 and 12, select the drive address rather than the unoccupied slot address.
If you want to perform a more complete test and your device does not have an autoloader, mount a writable scratch cartridge manually into the drive.
3. Enter *ntutil*.
4. Select **1** (Manual test).
5. Enter **50** (poll registered devices). Reply with **0** (zero) to the *Drive=0, Library=1* prompt. All devices detected by the SCSI Adapters (that were attached and powered-on at system boot time) should be displayed.
6. Enter **20** (open).
7. Enter **1** (RW).
8. Enter **49** (inquiry). Reply with **0** (zero) to the *Drive=0, Library=1* prompt. This step concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.
9. Enter **39** (test unit ready) until no error occurs. Each time you are prompted with the *Drive=0, Library=1* prompt, reply with **0** (zero).
10. Enter **31** (rewind).
11. Enter **87** (Read/Write Test). Enter **Y** in response to the confirmation message.
12. Enter **33** (unload).
13. Enter **21** (close).
14. Enter **99** (return to main menu).
15. Enter **9** (Exit *ntutil*).

Autochanger Device Attachment Test - Windows NT only

The following procedure tests the attachment of an autochanger device to a Microsoft Windows system. The procedure assumes that your device is attached at *lb1.0.0.1* and that the tape device is attached at *tape0*. You also need a cartridge in at least one of the slots. If you are using Microsoft Windows 200x, you must stop RSM to do this test.

When the **Return to continue:** message is displayed or when you are told to enter information, press the **Enter** key after following the instruction. When the utility displays information, use the appropriate hardware manual to verify that the information is reasonable.

1. Open a Windows command shell window.

Verifying Proper Attachment of Your Devices

2. Enter *ntutil*.
3. Select **1** (Manual Test).
4. Enter **50** (poll registered devices). All devices detected by the SCSI adapters (that were attached and powered-on at system boot time) should be displayed. You should see *lb1.0.0.1* in the list of found devices.
5. Enter **8** (Library Mode).
6. Enter **20** (open).
7. Enter **1** (Read/Write).
8. Enter **49** (inquiry). Reply with **1** (one) to the *Drive=0, Library=1* prompt. This concludes a very basic test of the device, SCSI connection, and the device driver. You can stop the test here or continue to perform a more complete test.
9. Enter **39** (test unit ready) until no error occurs. Each time you are prompted with the *Drive=0, Library=1* prompt, reply with **1** (one).
10. Enter **12** (*ioctl* initialize element status).
11. Enter **10** (*ioctl* return library inventory).
12. From the output in the previous step, select a writable, scratch cartridge and determine its address. Also, select the address of an unoccupied slot.
13. Enter **11** (*ioctl* move medium), then supply the address of the cartridge and the address of the unoccupied slot. Verify that the cartridge moved.
14. Enter **10** (*ioctl* return library inventory). Verify that the inventory was updated properly.
15. Enter **21** (close).
16. Enter **99** (return to main menu).
17. Enter **9** (Exit *ntutil*).

Appendix C. Managing the Microcode on the IBM Tape Drive

Microcode is computer software that is stored in nonvolatile storage on your tape device or library hardware. It controls the operation of your hardware. When your tape device or library hardware was manufactured, a microcode load was installed and shipped with your device.

If you are having trouble with your hardware, IBM service personnel will ask you what level of microcode you have on your hardware. If they believe you need a new level of microcode, they might instruct you to install a newer level of microcode on your hardware. They will provide you with updated microcode.

You can query the current level of microcode by issuing commands on the front panel of your hardware. Consult the appropriate hardware reference manual for specific instructions on querying your microcode level.

You can also query the last four digits of the current level of microcode using software if your device is connected to a host system that has device or library support. The unit must be powered-on, configured properly, and ready. See the appropriate chapter in this book (based on the host system that your hardware is attached to) for details on how to have the device ready.

AIX	Use the <i>tapeutil</i> command with the <i>vpd</i> subcommand. See the Revision Level output field.
HP-UX	Use the <i>tapeutil -f drive -o qmc -v</i> command where <i>drive</i> is the device special file of the attached tape device.
Linux	Use the <i>IBMtapeutil -f drive inquiry</i> command where <i>drive</i> is the device special file of the attached tape device.
Sun Solaris	Use the <i>tapeutil -f drive -o qmc -v</i> command where <i>drive</i> is the device special file of the attached tape device.
Tru64	Use the <i>IBMrasutil -f device</i> command where <i>device</i> is the device special file of the attached tape or medium changer device.
Microsoft Windows	Use the <i>ntutil</i> command with the <i>inquiry</i> subcommand. See the Microcode Revision Level output field.

The following instructions guide you to install another version of microcode on your tape drive:

1. Ensure that the tape drive is connected to a host system and that the tape device driver is powered-on and configured properly with no tape cartridge in the drive. Follow the instructions in Appendix B, "Verifying Proper Attachment of Your Devices," on page 289 to ensure that the drive is configured properly and ready.
2. Follow the platform-specific instructions.

AIX System

1. Open an AIX window.
2. Enter *tapeutil*. A menu will be displayed.
3. Enter **4** (Tape Drive Service Aids).
4. Select **Microcode Load**.
5. Select the device special file identifier for the device on which to load the microcode, then press the **Enter** key.
6. Specify the special file from where the microcode image is to be read, then press **F7** (Commit) to load the microcode onto the drive. When the loading is complete, press **F10** (Exit) to return to *tapeutil*.
7. Enter **Q** (Quit Program).

Compaq Tru64 System

1. Bring up a shell prompt.
2. Enter *IBMrasutil*. It will prompt you to enter a device file name.
3. Enter */dev/device name* when prompted for the device name.
4. Enter **2** (Update MicroCode).
5. Specify the special file from which the microcode image is to be read, then press the **Enter** key.
6. Enter **3** (Quit).

HP-UX System

This procedure assumes that the */dev/rmt/0mn* device is being updated.

1. Bring up a shell prompt.
2. Enter *tapeutil*. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter */dev/rmt/0mn* when prompted for the device name.
5. Enter **1** (Read/Write).
6. Select **7** (DOWNLOAD MICROCODE option).
7. Specify the special file from which the microcode image is to be read, then press the **Enter** key.
8. Enter **Q** (Quit Program).

Linux System

This procedure assumes that the */dev/IBMtape0* tape device is being updated.

1. Open a Linux window.
2. Enter *IBMtapeutil*. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter */dev/IBMtape0* when prompted for the device name.
5. Enter **1** (Read/Write).
6. Select **72** (Load Ucode).
7. Specify the file name from which the microcode image is to be read, then press the **Enter** key.
8. Enter **Q** (Quit Program).

Sun Solaris System

This procedure assumes that the `/dev/rmt/0st` tape device is being updated.

1. Bring up a shell prompt.
2. Enter `tapeutil`. A menu will be displayed.
3. Enter **1** (Open a Device).
4. Enter `/dev/rmt/0st` when prompted for the device name.
5. Enter **1** (Read/Write).
6. Select **7** (Download Microcode).
7. Specify the special file from which the microcode image is to be read, then press the **Enter** key.
8. Enter **Q** (Quit Program).

Microsoft Windows System

This procedure assumes that the new microcode is stored in `c:\mydata\d0i9_430.fmr` and that the drive at `tape0` is being updated. The name of the file containing the microcode must have the `fmr` extension.

1. Open a Windows command shell window.
2. Change to the `c:\mydata` directory.
3. Enter `ntutil`.
4. Select **1** (Manual test).
5. Enter **20** (open).
6. Enter **1** (Read/Write).
7. Enter **39** (test unit ready) until a *not ready* error is reported.
8. Enter **82** (Update MicroCode).
9. Enter `d0i9_430`.
10. Enter **21** (close).
11. Enter **99** (return to main menu).
12. Enter **9** (Exit `ntutil`).

Managing Microcode on the IBM Tape Drive

Appendix D. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries (or regions) in which IBM operates.

Any references to an IBM program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product 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. Evaluation and verification of operation in conjunction with other products, except those expressly designed by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country (or region) where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states (or regions) do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries (or regions), or both:

AIX	pSeries
AIX 5L	RS/6000
@server	StorageSmart
IBM	StorWatch
IBMLink	Tivoli®
Magstar®	TotalStorage
Netfinity®	

Intel and Pentium are registered trademarks of Intel Corporation in the United States, other countries (or regions), or both.

Microsoft, Windows, Windows NT, Windows 2000, Windows Server 2003, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries (or regions), or both.

UNIX is a registered trademark of The Open Group in the United States and other countries (or regions).

The following terms are trademarks of Hewlett-Packard, IBM, and Seagate in the United States:

Linear Tape-Open	LTO	Ultrium
------------------	-----	---------

Other company, product, and service names may be the trademarks or service marks of others in the United States, other countries (or regions), or both.

Index

Numerics

3494 Library Emulation Support for SCSI
Medium Changers 281

A

AIX

Alternate Pathing and Failover Support
 configuring and unconfiguring primary and alternate devices 34
 querying primary and alternate path configuration 34
Alternate Pathing Support 31
 Automatic Failover 31
 configuring and unconfiguring alternate pathing support 33
 Dynamic Load Balancing 32
 primary and alternate paths 33
Installation and Configuration Instructions
 preinstallation considerations 19
Tape Drive Service Aids
 Detailed Description 47
 error log analysis 49
 force microcode dump 47
 microcode load 48
 read dump 48
 reset drive 49

AIX (Atape)

Alternate Pathing Support for the IBM 3584 Library 27
Device and Volume Information
 Logging 53
 log file 53
 tape log utility 53
Installation and Configuration Instructions 19
 configuring tape and medium changer devices 20
 deconfiguring medium changer devices 21
 deconfiguring tape devices 21
 installation procedure 19
 uninstall procedure 21
Performance Considerations 51
 Before Calling Support 52
 common utilities 51
 data path 51
Problem Determination 55
 ATRC utility 57
 automatic dump facility (ADF) for the 3580 Ultrium tape drive 57
 detail data 55
 error logging 55
 trace facility 57
Product Requirements 11
 Data Flow 11
 Hardware Requirements 12

AIX (Atape) (continued)

 Product Requirements (continued)
 Software Requirements 11
 Special Files 23
 Special Files for medium changer devices 25
 Special Files for tape devices 23
Tape Drive Service Aids 47
Tape Drive, Media, and Device Driver Parameters 15
 autoloading 15
 block size 16
 compression 16
 configuration parameters 15
 logging 16
 maximum size of log file 16
 media parameter 17
 record space mode 16
 rewind immediate 17
 trailer labels 17
Tape Utility Program (tapeutil) 37
 Block Size and SCSI Transfer Length 45
 command-line interface 37
 Configuration Parameters 45
 General Subcommands 38
 interactive interface 37
 Medium Changer Subcommands 40
 Reserve and Release Commands 46
 Service Aid Subcommands 45
 tape drive service aids 46
 Tape Subcommands 41
 volume ID for logging 46
Using Dump Support 35
AIX device driver 3
AIX device parameters 51
AIX managing microcode 298
AIX medium changer device attachment test 290
AIX system, attachment test 289
AIX tape device attachment test 289
alternate pathing support
 library control path failover 97

C

changes, summary of xvii
Comments xv
CPF commands 120

D

device driver
 IBM SCSI tape/medium changer device driver/Solaris 127, 177
 IBM tape device driver/AIX enhanced 11

device driver (continued)

 IBM tape device driver/Compaq Tru64 61
 IBM tape device driver/HP-UX 69, 221

F

feature codes
 CPF and DPF 178

H

HP device driver 3
HP-UX (ATDD)
 Configuration Parameter Definitions 105
 device-specific parameters 105
 driver-specific parameters 106
Install, Uninstall and Configure 75, 85
Install, Uninstall, and Configuration adding an Ultrium device using current atdd driver 93
configure medium changers using kernel configuration 91
configure selected medium changers using ATDD header file 92
copy software to software depot 79, 89
create device-specific configuration entries (optional) 78, 88
create drive configuration file (optional) 76, 86
create hardware path entry 77, 87
determine drive hardware path for 3581 76, 86
determine drive hardware paths for 3580 and 3583 with fibre channel attachment 77, 87
determine drive hardware paths for 3583/3584 Tape Libraries with SCSI attachment 77, 87
install ATDD 81, 90
Install drivers manually 79, 88
Install using CD installation script 78
other administrative tasks 94
post configuration if medium changer devices 91
Power off tape drives 78, 88
review ATDD README file 80, 90
uninstalling the software 94
Introduction and Product Requirements 69
 Product Requirements 69
 Purpose 69

- HP-UX (ATDD) *(continued)*
 - Product Requirements
 - ATDD implementation 69
 - data flow 73
 - hardware requirements 69
 - software interface 74
 - software requirements 72
 - Special Files 101
 - Supported Configurations 103
 - modifying configuration parameters 103
 - Tape Utility Program (tapeutil) 113
 - copy software to software depot 114
 - hardware requirements 113
 - install manually 114
 - install the product 115
 - install using CD installation script 114
 - other administrative tasks 116
 - review tapeutil README file 115
 - tapeutil command-line interface 119
 - tapeutil implementation 113
 - tapeutil interactive interface 117
 - uninstalling the software 116
 - Using tapeutil 116
 - tapeutil service commands 119
 - Troubleshooting 107
 - Atdd_d log daemon 108
 - error logging 107
 - Problems and Solutions 110
 - Support_info script 107
 - tracing facility 107
- HP-UX (ATDD) basic SCSI commands 119
- HP-UX (ATDD) flag description 122
- HP-UX (ATDD) medium changer commands 120
- HP-UX (ATDD) tape drive commands 120
- HP-UX autochanger device attachment test 292
- HP-UX managing microcode 298
- HP-UX system, attachment test 291
- HP-UX tape device attachment test 291

I

- IBM device driver
 - SCSI tape/medium changer device driver/Solaris 127, 177
 - tape device driver/AIX enhanced 11
 - tape device driver/Compaq Tru64 61
 - tape device driver/HP-UX 69, 221

L

- libsmc extension 281
- Linux (IBMtape)
 - Tape Drive Service Aids
 - Detailed Description 173
 - force drive dump 173
 - load microcode 173
 - read dump 173
 - reset drive 173

- Linux (IBMtape) *(continued)*
 - Tape Utility Program (IBMtapeutil)
 - Service Aid Subcommands 170
 - Linux allow subcommand 164, 168
 - Linux append subcommand 164
 - Linux asf [Count] subcommand 164
 - Linux attachment verification 292
 - Linux audit [Address [Count] subcommand 169
 - Linux Automatic Cartridge Facility Mode 171
 - Linux Block Size and SCSI Transfer Size 171
 - Linux bsf [Count] subcommand 164
 - Linux bsfm [Count] subcommand 164
 - Linux bsr [Count] subcommand 164
 - Linux Capacity Scaling 171
 - Linux Compress subcommand 164
 - Linux Configuration Parameters 171
 - Linux Configuring Tape and Medium Changer Devices 140
 - Linux Create Special Files 172
 - Linux Density subcommand 165
 - Linux device driver 3
 - Linux Device Driver (IBMtape)
 - Alternate Pathing Support for Library Control Path Failover 153
 - Installation and Configuration Instructions 137
 - configuring tape and medium changer devices 140
 - installation procedure 138
 - querying installed package 139
 - update procedure 139
 - verifying/updating installation 139
 - Introduction and Product Requirements 127
 - Product Requirements
 - data flow 127
 - hardware requirements 128
 - software requirements 127
 - Tape Drive, Media, and Device Driver Parameters 131
 - Changeable parameters 133
 - configuration parameters 131
 - Nonchangeable parameters 132
 - Linux devids subcommand 169
 - Linux elementinfo subcommand 169
 - Linux eof [Count] subcommand 165
 - Linux Erase subcommand 165
 - Linux exchange Source Dest1 Dest2 subcommand 169
 - Linux fsf [Count] subcommand 165
 - Linux fsm [Count] subcommand 165
 - Linux fsr [Count] subcommand 165
 - Linux IBMtapeutil Command-Line Mode 161
 - Linux IBMtapeutil General Subcommands 162
 - Linux IBMtapeutil Interactive Mode 161
 - Linux IBMtapeutil Medium Changer Subcommands 168
 - Linux IBMtapeutil Tape Subcommands 164
 - Linux inquiry [Page] 163
 - Linux inventory subcommand 169

- Linux list subcommand 165
- Linux load subcommand 165
- Linux lock subcommand 165
- Linux Logical Write Protect 171
- Linux logpage "Page" 163
- Linux managing microcode 298
- Linux medium changer device attachment test 293
- Linux modepage "Page" 163
- Linux mount [Slot] subcommand 169
- Linux move Source Destination subcommand 170
- Linux mtdevice subcommand 165
- Linux Nocompress subcommand 164
- Linux offline subcommand 165
- Linux parms subcommand 166
- Linux position Destination subcommand 170
- Linux prevent subcommand 166, 170
- Linux print "Text" 163
- Linux qryinquiry subcommand 166
- Linux qrypos subcommand 166
- Linux qrysense subcommand 166
- Linux qryversion 163
- Linux read -d Destination [-c Count] subcommand 166
- Linux release 164
- Linux Release command 171
- Linux reqsense 164
- Linux reserve 164
- Linux Reserve command 171
- Linux retension subcommand 166
- Linux rewind subcommand 166
- Linux rtest [-b Blocksize] [-c Count] [-r Repetition]
 - Linux rtest [-b Blocksize] [-c Count] [-r Repetition] Tape Subcommand 166
- Linux rwtest [-b Blocksize] [-c Count] [-r Repetition] subcommand 167
- Linux seod subcommand 167
- Linux setblk [Block Size] subcommand 167
- Linux setpos [Blockid] subcommand 167
- Linux Special Files for Tape Devices 151
- Linux Special Files for the Medium Changer Device 151
- Linux status subcommand 168
- Linux sync subcommand 168
- Linux tape device attachment test 292
- Linux tape drive service aids 173
- Linux Tape Drive Service Aids 172
- Linux tape utility program (IBMtapeutil) 161
- Linux tell subcommand 168
- Linux tur 164
- Linux Uninstall Procedure 143
- Linux unlock subcommand 168
- Linux unmount [Slot] subcommand 170
- Linux weof [Count] subcommand 165
- Linux write -s Source subcommand 168
- Linux wtest [-b Blocksize] [-c Count] [-r Repetition] subcommand 168

M

- Managing microcode 297
- Microsoft Windows managing microcode 299
- Microsoft Windows NT autochanger device attachment test 295
- Microsoft Windows system, attachment test 295
- Microsoft Windows tape device attachment test 295
- mtlib command 281

N

- Notices 301

O

- Online access 287

R

- Related Information xiii
 - Additional Information xiv
 - IBM TotalStorage UltraScalable Tape Library 3584 xiii
 - IBM TotalStorage Ultrium External Tape Drive 3580 xiii
 - IBM TotalStorage Ultrium Scalable Tape Library 3583 xiii
 - IBM TotalStorage Ultrium Tape Autoloader 3581 xiii
 - IBM TotalStorage Ultrium Tape Library 3582 xiii
- rewoffl, rewoffl subcommand 165
- rewoffl, unload subcommand 165

S

- software requirements
 - CPF and DPF feature codes 178
- Solaris autochanger device attachment test 294
- Solaris device driver 3
- Solaris Device Driver (IBMtape)
 - Installation, Removal and Configuration 181
 - Adding or removing devices 191
 - Configuration parameters 188
 - Installation steps 182
 - Installing or updating IBMtape 182
 - Preventing conflicts 181
 - Removing IBMtape 188
 - Introduction and Product Requirements 177
 - Data flow 180
 - Hardware requirements 177
 - Product Requirements and Capability 177
 - Purpose 177
 - Software interface 180
 - Software requirements 178
 - Service and Diagnostic Aids 207
 - Downloading microcode 208

- Solaris Device Driver (IBMtape)
 - (continued)
 - Service and Diagnostic Aids (continued)
 - Forcing and storing device diagnostic dump 208
 - Functional verification 207
 - Problem determination 207
 - Setting the IBM_trace level 211
 - Tape and Medium Changer Utility Program (tapeutil) 212
 - Tracing facility 210
 - Special Files 193
- Solaris managing microcode 299
- Solaris system, attachment test 294
- Solaris tape device attachment test 294
- StorageSmart by IBM Ultrium device drivers
 - Attachment to open systems servers
 - Ultrium Products 7
- summary of changes xvii

T

- Trademarks 302
- Tru64
 - Installation and Configuration Instructions 63
 - Command-Line mode 64
 - Command-Line options 65
 - Installation procedure from the device driver CD 63
 - Installation procedure from the device driver ftp site 63
 - Interactive mode 64
 - Query procedure 64
 - uninstall procedure 64
 - Update procedure 64
 - Verify procedure 64
 - Introduction and Product Requirements 61
 - Hardware Requirements 61
 - Product Requirements 61
 - Purpose 61
 - Setting up the environment 61
 - Software Requirements 61

U

- Ultrium device driver platform support 3
- Ultrium device drivers 3
 - Attachment to open systems servers 4
 - 3580 Ultrium Tape Drive 6
 - 3581 Ultrium Tape 2U Autoloader 6
 - 3581 Ultrium Tape Autoloader 6
 - 3582 Ultrium Tape Library 6
 - 3583 Ultrium Scalable Tape Library 7
 - 3584 UltraScalable Tape Library 7

V

- Verifying device attachment 289

W

- Windows 200x Device Driver Management (IBMtape) 239
 - Disable Procedure 252
 - Installation Overview 239
 - Installation Procedures 239
 - Removal Procedure 252
 - uninstalling the device drivers 253
- Windows device driver 3
- Windows Device Drivers (IBMtape)
 - Introduction and Product Requirements 221
 - Windows 200x 224
 - hardware requirements 224
 - installation notes 225
 - software requirements 225
 - Windows NT 221
 - hardware requirements 221
 - installation notes 223
 - software requirements 223
- Windows NT Device Driver Management (IBMtape) 227
 - installation overview 227
 - installation procedure 227
 - Manual Starting and Stopping Procedures 236
 - removal procedure 233
- Windows Utility Program (ntutil) 257
 - Batch Mode 260
 - close 261
 - Command Statements 261
 - Comments 261
 - device_info 261
 - display_block (data) 261
 - dump_blocks (read and display blocks) 261
 - erase 262
 - find_devices 262
 - flush (buffer to media) 262
 - force_dump 262
 - forward_filemark 263
 - forward_record 263
 - get_last_sense 263
 - get_version 263
 - init_element_status 263
 - inquiry 263
 - load 264
 - locate_block_id 264
 - log_sense page_code 264
 - medium_info 264
 - move_medium 264
 - open 265
 - poll_devices 266
 - read 266
 - read_block_id 266
 - read_dump 266
 - read_write (test) 267
 - restore_data 267
 - return_error 267
 - return_lib_inventory_all 267
 - reverse_filemark 268
 - reverse_record 268
 - rewind 268
 - set_device_parameters (compression) 269
 - set_media_parameters (block size) 269

Windows Utility Program (ntutil)
(continued)
Batch Mode (continued)
 space_eod 269
 Supported Command Text
 Fields 261
 Supported Result Text Fields 270
 test_unit_ready 269
 unload 269
 update_code 269
 write 270
 write_filemark 270
Calling ntutil 257
Device Driver Diagnosis
 Information 275
Interactive Mode 258
Supported Result Text Fields
 block_id 271
 block_size 271
 delay 272
 exit 272
 exit_on_unexpected_result 271
 pause 272
 return_error_when_fail 271
 set 271
 symbols 272
 system 272
 type 271
Symbolic Values 272

Readers' Comments — We'd Like to Hear from You

IBM Ultrium Device Drivers
Installation and User's Guide

Publication No. GA32-0430-11

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department GZW
9000 South Rita Road
Tucson, Arizona U.S.A. 85744-0001



Fold and Tape

Please do not staple

Fold and Tape



Printed in USA

GA32-0430-11



Spine information:



IBM Ultrium Device Drivers

IBM Ultrium Tape Device Drivers: Installation and
User's Guide