

IBM TS4300 Tape Library



Tables of Contents

IBM TS4300 Tape Library	1
Overview	1
What's New in IBM TS4300 Tape Library	2
Introduction	2
Structure and supported library configurations	3
Components	7
Front panel	7
Rear panel	7
Magazines	8
Accessor	9
Power supply	10
User interfaces	10
Supported tape drives	10
Control path drives	12
Mixed drives	12
Drive sled back panels	12
Physical and logical addresses of drives	13
Drive features	14
Media optimization	14
Recommended access order (RAO) open function	15
Archive mode unthread	15
Speed matching	15
Channel calibration	15
Data cartridge capacity scaling	16
Power management	16
Encryption	16
Supported tape cartridges	16
Library functions	17
Encryption	17
Library sharing	17
Control path failover, Data path failover, and load balancing	18
Alerts and logging	18
Random and Sequential Logical Library modes	19
Planning	19
Acclimation	20
Library Layout and Location requirements	20
Power cords	23
Network requirements	26
Host requirements	27
SAS interface	27
Fibre Channel interface	27
Compatible Configurations	27
HBA requirements	28
Application software	28
Optional features	28
Installing	29
Unpacking the Base Module and Expansion Modules	30
Identifying Library Module components	31
Installing a tabletop module	31
Removing inner foam from base module	32
Preparing top and bottom modules	33
Installing modules in a rack	35
Aligning and connecting modules	37
Installing a tape drive	39
Connecting cables	39
Powering on the library	41
The Initial Setup process	41
Initial configuration and customization	42
Labeling and loading tape cartridges	42
Verifying the installation	43
Advanced library configuration	43
Overview	43
Library partitioning	44
Verifying the host connection	44
Registering for support notification	45
Managing	45
The Management GUI	45
The Operator Panel	47
Locating Management functions	48
Default settings	49
Methods of cleaning drives	51
Accessing cartridges	51

Configuring Library Managed Encryption	52
Troubleshooting	53
How the library reports problems	53
Identifying a failed component	54
Running library tests	54
Troubleshooting Guide	54
Pre-call checklist	57
Contacting IBM technical support	57
Diagnostic information	58
IBM Tape Diagnostic tool (ITDT)	58
Event codes	59
Main error events	59
Warning error events	64
Configuration Change events	67
Informational events	68
TapeAlert Flags	69
TapeAlert flags supported by the library	69
TapeAlert flags supported by the drive	70
Sense data	73
Drive Error Codes: Single-character display (SCD)	73
SCD dot	74
Status light	74
Upgrading and servicing	74
Internal view of library	75
Adding, removing, or replacing a tape drive	75
Adding or replacing a Base or Expansion Module	77
Adding, removing, or replacing a power supply	81
Replacing a Base or Expansion controller card	82
Installing, removing, or replacing an accessor and spooling mechanism	84
Returning the accessor to the Base Module	88
Removing or replacing a spooling mechanism	89
Removing or replacing a magazine	91
Moving the library modules	92
Reference	92
Minimum firmware levels for common library features	93
Management GUI functions and roles	93
TS4300 Help pages	96
REST API for scalable tape libraries	96
Library Configuration Forms	96
Library information	97
Module and drive information	97
Logical Library information	98
LTO media	98
Data cartridges	99
Cartridge Read/Write compatibility	100
LTO type M cartridge (M8)	100
Capacity scaling	100
WORM (Write once, read many) cartridges	101
WORM media	101
Data security on WORM media	101
WORM media errors	101
Requirements for WORM capability	101
Cleaning cartridge	101
Cartridge memory chip (LTO-CM)	102
Bar code labels	102
Guidelines for the use of bar code labels	103
Write-Protect switch	103
Cartridge care and handling	104
Provide training	104
Ensuring proper packaging	104
Proper acclimation and environmental conditions	104
Completing a thorough inspection	104
Handling the cartridge carefully	105
Examples of cartridge problems	105
Repositioning or reattaching a leader pin	105
Repositioning a leader pin	106
Reattaching a leader pin	106
Environmental and shipping specifications for tape cartridges	108
Disposing of tape cartridges	109
Ordering media supplies	109
Ordering bar code labels	110
Replacement parts	110
Manual cartridge removal procedure	111
Recommended tools	112

Before you begin	112
Beginning procedure	112
Removing the drive brick from the sled	112
Removing the drive cover	113
Full height drive: Tape spooled off supply reel	114
Half height drive: Tape spooled off supply reel	115
Full height drive: Tape pulled from or broken near leader pin	115
Half height drive: Tape pulled from or broken near leader pin	116
Full height drive: Tape broken in mid-tape	117
Half height drive: Tape broken in mid-tape	117
Full height drive: Tape tangled along tape path	118
Half height drive: Tape tangled along tape path	119
Full height drive: No apparent failure or damage to tape	120
Half height drive: No apparent failure or damage to tape	121
Ending procedure	122
Accessibility	123
Notices	123
Trademarks	124
Terms and conditions for product documentation	125
Homologation statement	125
Electromagnetic compatibility notices	126
Canada Notice	126
European Community and Morocco Notice	126
Germany Notice	126
Japan Electronics and Information Technology Industries Association (JEITA) Notice	127
Japan Voluntary Control Council for Interference (VCCI) Notice	127
Korea Notice	127
People's Republic of China Notice	127
Russia Notice	127
Taiwan Notice	128
United Kingdom Notice	128
United States Federal Communications Commission (FCC) Notice	128
Safety and environmental notices	128
Safety notices	128
Possible safety hazards	129
Laser safety and compliance	130
Performing the safety inspection procedure	130
Rack safety	130
Power cords	132
Cautions and regulatory compliance statements for NEBS	132
Glossary	132
Publications	144
Related publications	145
IBM Tape Device Drivers and Diagnostic Tool User's Guide	145
Introduction	146
Common extended features	147
Path failover and load balancing	147
Supported devices and feature codes	147
Automatic failover	148
Data path failover	148
Control path failover	149
Dynamic load balancing	150
Dynamic Runtime Attributes	150
Data encryption	150
Tape and library requirements	150
Planning for application-managed tape encryption	151
Planning for library-managed tape encryption	151
Encryption feature codes	152
Recommended access order (RAO) open function	152
AIX Tape and Medium Changer device driver	152
Product requirements	153
Installation and configuration instructions	153
Installation procedure	154
Configuring Tape and Medium Changer devices	154
Configuring limitations	154
Deconfiguring tape devices	154
Deconfiguring Medium Changer devices	155
Uninstalling	155
Tape drive, media, and device driver parameters	155
Configuration parameters	155
Media parameters	158
Special files	159
Special files for tape devices	159

Special files for Medium Changer devices	160
Persistent Naming Support	160
Changing the logical name after initial boot	161
Control Path failover support for tape libraries	161
Configuring and unconfiguring path failover support	161
Primary and alternative paths	162
Querying primary and alternative path configurations	162
Configuring and unconfiguring primary and alternative devices	162
Data Path failover and load balancing support for tape drives	162
Installing Data Path failover license key	163
Configuring and unconfiguring path failover support	163
Primary and alternative paths	163
Querying primary and alternative path configuration	163
Configuring and unconfiguring primary and alternative devices	163
System-managed encryption	164
Device driver configuration	164
Querying tape drive configuration	164
Testing data encryption configuration and connectivity	164
Error logging	164
Field support information	164
Problem determination	165
Dump support	165
Device and volume information logging	165
Log file	165
Tape log utility	166
Reservation conflict logging	166
Error logging	166
Error log templates	166
Automatic dump facility	168
Dynamic Runtime Attributes	168
Trace facility	168
Atape System Trace (ATRC) utility	168
Component tracing	169
Atape Component Trace (ACTRC) utility	169
Tape drive service aids	169
Tape drive service aids details	169
Performance considerations	170
Data path	171
Common AIX utilities	171
AIX iostat utility for tape performance	171
Before Support is called	171
Linux Tape and Medium Changer device driver	171
Product requirements	172
Installation and Configuration instructions	172
Conventions used	172
Configuration limitations	172
Components created during installation	172
Installation procedure	173
Updating procedure	173
Querying the installed package	174
Configuring Tape and Medium Changer devices on Intel-compatible systems	174
Configuring Tape and Medium Changer devices on IBM System p models	174
Configuring Tape and Medium Changer devices on IBM System z models	174
Uninstallation procedure	175
Tape drive, media, and device driver parameters	175
Configuration parameters	175
Nonchangeable parameters	176
Changeable parameters	177
Special files	179
Special files for the tape device	179
Special files for the Medium Changer device	179
Persistent naming support	179
Control Path failover support for tape libraries	180
Configuring and unconfiguring path failover support	181
Enabling path failover for the sg interface	181
Primary and alternative paths	182
Querying primary and alternative path configuration	182
Disabling and enabling primary and alternative paths	182
Data Path failover and load balancing support for tape drives	182
Enabling path failover for st and sg interface	183
Primary and alternative paths	183
Querying primary and alternative path configuration	184
Disabling and enabling primary and alternative paths	184

Tape Reserve Type	184
Open source device driver - lin_tape	184
Comparing IBMtape and lin_tape	185
Installation	185
Driver parameters and special device files	185
Taking devices offline and completing maintenance	185
Path failover support	186
lin_taped daemon	186
System-managed encryption	186
Configuring device drivers	186
Querying tape drive configuration	187
Problem determination	187
Tracing driver modules	187
Configuring and running the lin_taped daemon	187
Reservation conflict logging	189
Devices not reported at /proc/scsi/IBMchanger or /proc/scsi/IBMtape	190
Windows Tape and Medium Changer device driver	190
Product requirements	190
Installation and configuration instructions	191
Windows Server 2008, 2012, 2016, and 2019 instructions	191
Configuring limitations	193
Persistent Naming Support on Windows Server 2008, 2012, 2016, and 2019	193
Control Path failover support for tape libraries	193
Configuring and unconfiguring Control Path failover support	193
Querying primary and alternative path configuration	194
Checking disablement of Control Path failover setting	194
Data Path failover support for tape drives	194
Configuring and unconfiguring Data Path failover support	194
Reserve Type if DPF is disabled	194
Querying primary and alternative path configuration	194
Checking disablement of Data Path failover setting	194
System-managed encryption	195
Device driver configuration	195
Configuration file	195
Querying tape drive configuration settings	196
Problem determination	196
Windows Server 2008, 2012, 2016, and 2019 instructions	196
Reservation conflict logging	196
Max retry busy	197
Checking if Data Path failover is enabled	197
Checking if Control Path failover is enabled	197
Old releases: V6.2.5.2 and prior releases	197
IBM Tape Diagnostic Tool (ITDT)	198
Purpose	198
Accessing ITDT	198
Supported systems	199
IBM Tape Diagnostic Tool - Standard Edition	199
Installing ITDT - Standard Edition	199
Installing ITDT-SE on AIX operating systems	199
Installing ITDT-SE on Microsoft Windows operating systems	200
Installing ITDT-SE on i5/OS operating systems	200
Installing ITDT-SE on other supported operating systems	201
Starting ITDT - Standard Edition	202
Starting ITDT-SE on Solaris operating systems	202
Starting ITDT-SE on Windows operating systems	203
Starting ITDT-SE on i5/OS operating systems	203
Starting ITDT-SE on other supported operating systems	204
Generic Operating System driver with ITDT-SE	204
Standard Edition - known issues and limitations	204
AIX operating systems	204
HP-UX operating systems	204
Linux operating systems	205
Solaris operating systems	205
Windows operating systems	205
i5/OS operating systems	205
All supported operating systems	206
Standard Edition - Start menu commands	206
Standard Edition - Scan menu commands	207
Scan	208
Health Test	209
Dump	210
Firmware Update	211
System Test	212

Eject Cartridge	213
Cleaning Statistics	213
Other Functions	213
Full Write	213
Tape Usage	214
Check LTFS Readiness	215
Library Test	215
Library Media Screening	215
Encryption	215
Configure TCP/IP	216
Return	217
Standard Edition - Tapeutil menu commands	217
[1] Open a Device	218
[2] Close a Device	219
[3] Inquiry	219
[4] Test Unit Ready	219
[5] Reserve Device	219
[6] Release Device	219
[7] Request Sense	219
[8] Log Sense	219
[9] Mode Sense	219
[10] Query Driver Ver. (Version)	219
[11] Display All Paths	220
[12] Query Runtime Info	220
[20] Rewind	220
[21] Forward Space File Marks	220
[22] Backward Space File Marks	220
[23] Forward Space Records	220
[24] Backward Space Records	220
[25] Space to End of Data	220
[26] Read and Write Tests	220
[27] Read or Write Files	221
[28] Erase	221
[29] Load Tape	221
[30] Unload Tape	221
[31] Write File Marks	221
[32] Synchronize Buffers	221
[33] Query/Set Parameter	221
[34] Query/Set Tape Position	222
[35] Query Encryption Status	222
[36] Display Message	222
[37] Report Density Supp (Support)	222
[38] Test Encryp. Path (Test Encryption Key Path/Setup)	222
[39] Configure TCP/IP Port	222
[50] Element Information	223
[51] Position to Element	223
[52] Element Inventory	223
[53] Exchange Medium	224
[54] Move Medium	224
[55] Initialize Element Status	224
[56] Prevent/Allow Medium Removal	224
[57] Initialize Element Status Range	224
[58] Read Device IDs	224
[59] Read Cartridge Location	224
[70] Dump/Force Dump/Dump	225
[71] Firmware Update	225
[101] HDP Discover	225
[102] HDP Initiate Call Home	225
[103] HDP Show Import Export Elements	225
Add device manually	226
Standard Edition - Program options	226
Standard Edition - Tapeutil scripting commands	226
allow	229
devinfo	229
inquiry inq inqj	229
logpage logpagej	230
loop	230
modepage	230
prevent	231
print	231
qrypath	231
qryversion	231

release	231
reqsense	231
reserve	232
scan scanj	232
sleep	233
tur	233
vpd	233
append	234
bsf	234
bsr	234
channelcalibration	234
chgpart	234
density	234
display	234
erase	235
formattape	235
fdp	235
fdpl	235
fsf	235
fsr	236
getparms	236
idp	236
idpl	236
list	236
load	237
logsense	237
qrypar/qrypart	237
qrylbp	237
qrypos	237
qrytcpip	237
qrytemp	238
read	238
readattr	238
resetdrive	238
rmp	238
runtimeinfo/qryruntimeinfo	238
rewind	239
rtest	239
rwtest	239
sdp	239
sdpl	239
seod	240
setparm	240
setpos	240
settcpip	241
sync	241
unload	241
verlbp	241
weof	242
write	242
writeattr	242
wtest	242
audit	242
cartridgelocation	242
elementinfo	243
exchange	243
inventory	243
move	243
position	243
dump	243
ekmtest	244
encryption	244
ucode	244
tapephpcp	244
ltfsphpcp	244
verify	245
devicestatistics	245
checkltfsreadiness	245
ltfsdefragmentation	245
standardtest	246
fullwrite	246
systemtest	246

tapeusage	246
hdp discover	246
hdp senderror	247
hdp show	247
TS4500 REST over SCSI	247
TS4300: RESTful API	248
Deprecated commands	249
Alternative mount/demount script - sample for Windows	249
Standard Edition scripting commands: known limitations and deviations	251
IBM Tape Diagnostic Tool - Graphical Edition	251
Installing ITDT - Graphical Edition	251
Installing ITDT-GE on Windows operating systems	251
Installing ITDT-GE on Linux operating systems	252
Graphical Edition - known issues and limitations	252
Linux operating systems	252
Windows operating systems	252
All supported operating systems	252
ITDT-GE user interface description	253
Graphical Edition - Scan menu commands	254
Scan	255
Health Test	256
Configure TCP/IP Ports	257
Dump	257
Firmware Update	257
Firmware Update - check for updates	258
Encryption	258
Full Write	259
Tape Usage	260
System Test	261
Library Diagnostic Self-Test	261
Library Media Screening	261
LTFS Readiness Check	261
Manual Inspection Record Entry	262
Graphical Edition - Copy Services	262
Graphical Edition - visualizing log files	263
Graphical Edition - Tapeutil menu commands	263
Open	266
Close	266
Inquiry	267
Test Unit Ready	267
Reserve Device	267
Release Device	267
Request Sense	267
Log Sense	267
Mode Sense	267
Query Driver Version	267
Display All Paths	267
Rewind	268
Forward Space Filemarks	268
Backward Space Filemarks	268
Servo Channel Calibration	268
Forward Space Records	268
Backward Space Records	268
Space to End of Data	268
Read and Write Tests	268
Read or Write Files	269
Erase	270
Load Tape	270
Unload Tape	270
Write Filemarks	270
Synchronize Buffers	270
Query/Set Parameter	270
Query/Set Position	270
Query Encryption Status	271
Display Message	271
Display All Paths	267
Report Density Support	271
Test Encryption Path	271
Element Information	271
Position to Element	272
Element Inventory	272
Exchange Medium	272

Move Medium	272
Initialize Element Status	272
Prevent/Allow Medium Removal	272
Initialize Element Status Range	272
Read Device IDs	273
Read Cartridge Location	273
Configure TCP/IP Ports	273
Dump/Force Dump/Dump	273
Firmware Update	273
Verifying correct attachment of your devices	274
Managing the microcode on the IBM tape drive	274
Notices	274
Terms and conditions for knowledge centers	275
IBM Tape Device Drivers Programming Reference	276
Introduction	277
Common extended features	277
Tape drive functions and device driver ioctls	278
Media partitioning	278
Data safe (append-only) mode	279
Read Position long/extended form and Locate(16) commands	279
Logical Block Protection	280
Programmable Early Warning (PEW)	280
Log Sense page and subpage	280
Mode Sense page and subpage	280
Verify Tape	280
RAO - Recommended Access Order	280
AIX tape and medium changer device driver	281
Software interface for tape devices	281
Software interface for medium changer devices	281
Special files	282
Special files for tape devices	282
Special files for medium changer devices	282
Opening the special file for I/O	283
The extended open operation	283
Writing to the special file	284
Reading from the special file	284
Reading with the TAPE_SHORT_READ extended parameter	284
Reading with the TAPE_READ_REVERSE extended parameter	284
Closing the special file	285
Device and volume information logging	285
Log file	285
Persistent reservation support and IOCTL operations	286
ODM attributes and configuring persistent reserve support	286
Default device driver host reservation key	286
Preempting and clearing another host reservation	287
Openx() extended parameters	287
AIX tape persistent reserve IOCTLs	287
Atape persistent reserve IOCTLs	289
General IOCTL operations	291
Overview	291
IOCINFO	292
STIOCMD	292
STPASSTHRU	293
SIOC_PASSTHRU_COMMAND	293
SIOC_INQUIRY	293
SIOC_REQSENSE	294
SIOC_RESERVE	295
SIOC_RELEASE	295
SIOC_TEST_UNIT_READY	295
SIOC_LOG_SENSE_PAGE	295
SIOC_LOG_SENSE10_PAGE	296
SIOC_MODE_SENSE_PAGE	297
SIOC_MODE_SENSE_SUBPAGE	297
SIOC_MODE_SELECT_PAGE	298
SIOC_MODE_SELECT_SUBPAGE	298
SIOC_QUERY_OPEN	298
SIOC_INQUIRY_PAGE	299
SIOC_DISABLE_PATH	299
SIOC_ENABLE_PATH	299
SIOC_SET_PATH	299
SIOC_DEVICE_PATHS	299
SIOC_QUERY_PATH	300
SIOC_RESET_PATH and SIOC_CHECK_PATH	301

SIOC_RESET_DEVICE	302
SIOC_DRIVER_INFO	302
Tape IOCTL operations	302
Overview	302
STIOCHGP	304
STIOCTOP	305
STIOCQRYP or STIOCSETP	306
STIOCSYNC	308
STIOCDM	309
STIOCQRYPOS or STIOCSETPOS	309
STIOCQRYSENSE	310
STIOCQRYINQUIRY	310
STIOC_LOG_SENSE	311
STIOC_RECOVER_BUFFER	312
STIOC_LOCATE	312
STIOC_READ_POSITION	312
STIOC_SET_VOLID	313
STIOC_DUMP	313
STIOC_FORCE_DUMP	313
STIOC_READ_DUMP	313
STIOC_LOAD_UCODE	314
STIOC_RESET_DRIVE	314
STIOC_FMR_TAPE	314
MTDEVICE (Obtain device number)	314
STIOC_PREVENT_MEDIUM_REMOVAL	315
STIOC_ALLOW_MEDIUM_REMOVAL	315
STIOC_REPORT_DENSITY_SUPPORT	315
STIOC_GET_DENSITY and STIOC_SET DENSITY	317
STIOC_CANCEL_ERASE	317
GET_ENCRYPTION_STATE	318
SET_ENCRYPTION_STATE	319
SET_DATA_KEY	319
READ_TAPE_POSITION	319
SET_TAPE_POSITION	321
SET_ACTIVE_PARTITION	321
QUERY_PARTITION	321
CREATE_PARTITION	322
ALLOW_DATA_OVERWRITE	323
QUERY_LOGICAL_BLOCK_PROTECTION	324
SET_LOGICAL_BLOCK_PROTECTION	324
STIOC_READ_ATTRIBUTE	325
STIOC_WRITE_ATTRIBUTE	326
VERIFY_TAPE_DATA	326
QUERY_RAO_INFO	327
GENERATE_RAO	327
RECEIVE_RAO	328
QUERY_ARCHIVE_MODE_UNTHREAD	329
SET_ARCHIVE_MODE_UNTHREAD	329
TAPE_LOAD_UNLOAD	330
GET_VHF_DEVICE_STATUS	330
Medium changer IOCTL operations	331
Overview	331
SMCIOE_ELEMENT_INFO	332
SMCIOE_MOVE_MEDIUM	333
SMCIOE_EXCHANGE_MEDIUM	333
SMCIOE_POS_TO_ELEM	333
SMCIOE_INIT_ELEM_STAT	334
SMCIOE_INIT_ELEM_STAT_RANGE	334
SMCIOE_INVENTORY	334
SMCIOE_LOAD_MEDIUM	336
SMCIOE_UNLOAD_MEDIUM	336
SMCIOE_PREVENT_MEDIUM_REMOVAL	336
SMCIOE_ALLOW_MEDIUM_REMOVAL	336
SMCIOE_READ_ELEMENT_DEVIDS	337
SMCIOE_READ_CARTIDGE_LOCATION	338
Return codes	339
Codes for all operations	340
Open error codes	340
Write error codes	340
Read error codes	341
Close error codes	341
IOCTL error codes	341

Linux tape and medium changer device driver	341
Software interface	341
Entry points	341
Medium changer devices	342
General IOCTL operations	343
Overview	343
SIOC_INQUIRY	344
SIOC_REQSENSE	345
SIOC_RESERVE	346
SIOC_RELEASE	346
SIOC_TEST_UNIT_READY	346
SIOC_LOG_SENSE_PAGE, SIOC_LOG_SENSE10_PAGE, and SIOC_ENH_LOG_SENSE	346
SIOC_MODE_SENSE_PAGE and SIOC_MODE_SENSE	348
SIOC_INQUIRY_PAGE	348
SCSI_PASS_THROUGH	349
SIOC_QUERY_PATH	349
SIOC_DEVICE_PATHS	350
SIOC_ENABLE_PATH	351
SIOC_DISABLE_PATH	351
Tape drive IOCTL operations	351
Overview	351
STIOCTOP	353
STIOCQRYP or STIOCSETP	353
STIOCSYNC	356
STIOCDM	356
STIOCQRYPOS	356
STIOCSETPOS	357
STIOCQRYSENSE	357
STIOCQRYINQUIRY	358
STIOC_LOCATE	359
STIOC_READ_POSITION	359
STIOC_RESET_DRIVE	359
STIOC_PREVENT_MEDIUM_REMOVAL	359
STIOC_ALLOW_MEDIUM_REMOVAL	360
STIOC_REPORT_DENSITY_SUPPORT	360
MTDEVICE (Obtain Device Number)	361
STIOC_GET_DENSITY and STIOC_SET_DENSITY	361
GET_ENCRYPTION_STATE	362
SET_ENCRYPTION_STATE	363
SET_DATA_KEY	363
STIOC_QUERY_PARTITION	364
STIOC_CREATE_PARTITION	364
STIOC_SET_ACTIVE_PARTITION	366
STIOC_ALLOW_DATA_OVERWRITE	366
STIOC_READ_POSITION_EX	368
STIOC_LOCATE_16	370
STIOC_QUERY_BLK_PROTECTION	370
STIOC_SET_BLK_PROTECTION	370
STIOC_VERIFY_TAPE_DATA	370
STIOC_QUERY_RAO	372
STIOC_GENERATE_RAO	372
STIOC_RECEIVE_RAO	373
STIOC_SET_SPDEV	374
Tape drive compatibility IOCTL operations	374
MTIOCTOP	375
MTIOCGET	375
MTIOCPOS	375
Medium changer IOCTL operations	375
SCSI IOCTL commands	375
SMCIOE_ELEMENT_INFO	376
SMCIOE_MOVE_MEDIUM	376
SMCIOE_EXCHANGE_MEDIUM	376
SMCIOE_POS_TO_ELEM	377
SMCIOE_INIT_ELEM_STAT	377
SMCIOE_INIT_ELEM_STAT_RANGE	377
SMCIOE_INVENTORY	378
SMCIOE_LOAD_MEDIUM	379
SMCIOE_UNLOAD_MEDIUM	379
SMCIOE_PREVENT_MEDIUM_REMOVAL	379
SMCIOE_ALLOW_MEDIUM_REMOVAL	379
SMCIOE_READ_ELEMENT_DEVIDS	380
Return codes	381

General error codes	381
Open error codes	381
Close error codes	382
Read error codes	382
Write error codes	382
IOCTL error codes	382
Windows tape device drivers	383
Windows programming interface	383
User-callable entry points	383
Tape Media Changer driver entry points	383
CreateFile	384
ReadFile	385
WriteFile	385
Write Tapemark	386
SetTapePosition	386
GetTapePosition	386
SetTapeParameters	387
GetTapeParameters	387
PrepareTape	387
EraseTape	388
GetTapeStatus	388
DeviceIoControl	388
Medium Changer IOCTLs	389
IOCTL commands	389
Preempt reservation	390
Vendor-specific (IBM) device IOCTLs for DeviceIoControl	391
IOCTL_TAPE_OBTAIN_SENSE	392
IOCTL_TAPE_OBTAIN_VERSION	392
IOCTL_TAPE_LOG_SELECT	393
IOCTL_TAPE_LOG_SENSE	393
IOCTL_TAPE_LOG_SENSE10	393
IOCTL_ENH_TAPE_LOG_SENSE10	394
IOCTL_TAPE_REPORT_MEDIA_DENSITY	394
IOCTL_TAPE_OBTAIN_MTDEVICE	394
IOCTL_TAPE_GET_DENSITY	395
IOCTL_TAPE_SET_DENSITY	395
IOCTL_TAPE_GET_ENCRYPTION_STATE	395
IOCTL_TAPE_SET_ENCRYPTION_STATE	396
IOCTL_TAPE_SET_DATA_KEY	396
IOCTL_CREATE_PARTITION	397
IOCTL_QUERY_PARTITION	397
IOCTL_SET_ACTIVE_PARTITION	398
IOCTL_QUERY_DATA_SAFE_MODE	398
IOCTL_SET_DATA_SAFE_MODE	398
IOCTL_ALLOW_DATA_OVERWRITE	399
IOCTL_READ_TAPE_POSITION	399
IOCTL_SET_TAPE_POSITION	400
IOCTL_QUERY_LBP	400
IOCTL_SET_LBP	401
IOCTL_SET_PEW_SIZE	401
IOCTL_QUERY_PEW_SIZE	402
IOCTL_VERIFY_TAPE_DATA	402
IOCTL_QUERY_RAO_INFO	402
IOCTL_GENERATE_RAO	403
IOCTL_RECEIVE_RAO	403
IOCTL_CHANGER_OBTAIN_SENSE	404
IOCTL_MODE_SENSE	404
Variable and fixed block read/write processing	405
Event log	405
Notices	407
How to send your comments	408
Terms and conditions for knowledge centers	408

IBM TS4300 Tape Library

Welcome to the IBM TS4300 tape library documentation, where you can find information about how to install, maintain, and use the TS4300 tape library.

Last updated: 2022-04-04

- [Overview](#)
The overview of the features and functions of the IBM® TS4300 tape library is useful for high-level evaluation of the product and planning for the implementation of the product.
- [Planning](#)
The library requires an environment able to accommodate the appropriate space, power, location, and other technical specifications. Use this section as a reference for onsite requirements to allow for optimum operation of the library.
- [Installing](#)
Use this section to follow the procedures to install and configure your library.
- [Managing](#)
Four user roles are described, and each user role has its specific functions.
- [Troubleshooting](#)
Use the information in this section to troubleshoot any issues with your library setup and configuration.
- [Upgrading and servicing](#)
In this section, you can follow the procedures to add, remove, and replace library components.
- [Reference](#)
This section provides information about the Help pages, REST API, LTO Media, and other reference documentation.
- [Notices](#)
This information was developed for products and services that are offered in the US. This material might be available from IBM in other languages. However, you might be required to own a copy of the product or product version in that language in order to access it.
- [Glossary](#)
This glossary defines the special terms, abbreviations, and acronyms that are used in this publication. If you do not find the term that you are looking for, refer to the index or to the *Dictionary of Computing*, 1994.
- [Publications](#)
The Publications section provides instructions to print selected IBM Documentation topics, and can help you locate other publications that are related to the libraries and drives.

Getting started

- [Overview](#)
- [Planning](#)
- [Installing](#)
- [Managing](#)

Common tasks

- [Upgrading and servicing](#)
- [Cartridge care and handling](#)
- [Methods of cleaning drives](#)
- [HBA requirements](#)

Troubleshooting and support

- [Troubleshooting](#)
-  [IBM tape storage support](#)
-  [IBM Support home page](#)

More information

- [Minimum firmware levels for common library features](#)
- [TS4300 Help pages](#)
- [REST API for scalable tape libraries](#)
- [Related publications](#)
-  [IBM TS4300 Tape Library SCSI Reference](#)
-  [IBM Community \(Community platform\)](#)
-  [IBM Support content \(Product support\)](#)
-  [IBM Spectrum Archive Single Drive Edition documentation](#)
-  [Redbooks home page](#)

©Copyright IBM Corporation 2017-2021. Last updated: 2022-04-04

Overview

The overview of the features and functions of the IBM® TS4300 tape library is useful for high-level evaluation of the product and planning for the implementation of the product.

- [What's New in IBM TS4300 Tape Library](#)

- [Introduction](#)
The IBM® TS4300 tape library provides compact, high-capacity, low-cost solutions for simple, unattended data backup. The installation begins with the 3U high Base Module, with capacity for up to 40 tape cartridges and 3 half-height LTO tape drives, or one full-height and one half-height tape drive.
- [Structure and supported library configurations](#)
The TS4300 library supports a single base module tabletop configuration or a scalable rackmount configuration.
- [User interfaces](#)
This library has two user interfaces: the Management GUI and the Operator Panel.
- [Supported tape drives](#)
The library supports LTO6 and later tape drives. Each drive requires a SAS or Fibre Channel cable.
- [Drive features](#)
- [Supported tape cartridges](#)
- [Library functions](#)
The library provides many specific functions, such as random or sequential operating mode, encryption, library sharing, path failover, and alerts and logging.

What's New in IBM TS4300 Tape Library

Read about information that is new or changed in this release of the IBM® TS4300 Tape Library documentation. Install server using configured settings.

New information

The following section contains updated information for the current edition.

- LTO Media
- Supported tape drives and supported tape cartridges
- Installing table feet
- TS4300 Help pages
- Event Codes
- [Replacement parts](#)

Introduction

The IBM® TS4300 tape library provides compact, high-capacity, low-cost solutions for simple, unattended data backup. The installation begins with the 3U high Base Module, with capacity for up to 40 tape cartridges and 3 half-height LTO tape drives, or one full-height and one half-height tape drive.

Each module type has its special designation.

Table 1. Module designations

MTM	Description
3555-L3A	Base Module
3555-E3A	Expansion Module

[Figure 1](#) shows a two-module version of the tape library. The library on the left shows the base module above the expansion module. The library on the right shows the base module below the expansion module. An individual library can consist of one base module and up to six expansion modules. See [Structure and supported library configurations](#) for supported configurations.

Figure 1. Two module tape library



The library provides the following capabilities:

- New user interface for improved usability
- Updated library communication system
- I/O magazines to allow individual cartridge handling to be done independent of the library.
- Mixed media types
- Integrated management tools
- Scalability to seven modules
- Remote management with the management GUI
- Remote monitoring with Simple Network Management Protocol (SNMP)
- Multipath architecture
- Host-based path failover
- Sequential Mode option

Table 2. Minimum and maximum storage configurations

Configuration	Capacity	
Minimum 1 Module Library Base Module only	Cartridges	40 ² 35 with 5-slot I/O ¹
	Half height ³ Tape Drives	3

Configuration	Capacity	
Maximum 7 Module Library Base Module 6 Expansion Modules	Full height ⁴ Tape Drives	1
	Half height ³ /Full height ⁴ Tape Drive Mix	1/1
	Cartridges	280 ² 275 with 5-slot I/O ¹
	Half height ³ Tape Drives	21
	Full height ⁴ Tape Drives	7
	Half height ³ /Full height ⁴ Tape Drive Mix	Nineteen half-height drives to one full-height drive, or Seven half-height drives to seven full-height drives
Remember:		
¹Every module can contain a 5-slot I/O station. If a seven module library has an I/O station for each module, the maximum number of slots that can be configured as I/O slots are 35. ²For libraries with serial numbers before 7800K0K, capacity in lowest module for a 5-slot I/O station is 32 slots and for a 4-slot I/O station is 28 slots. ³Half-height tape drives can be installed in any drive bay in a module. ⁴Full-height tape drives must be installed in the lowest two bays of a module. Installing a full-height drive in the top two bays of a module is not supported.		

Structure and supported library configurations

The TS4300 library supports a single base module tabletop configuration or a scalable rackmount configuration.

Supported library configurations

All libraries start with a Base Module. Up to six Expansion Modules can be added as needed to support customer requirements. The architecture is designed to support a maximum of three Expansion Modules above the Base Module and three Expansion Modules below the Base Module. [Table 1](#) shows the supported configurations for libraries, ranging 1 - 7 total modules.

Figure 1. Base Module

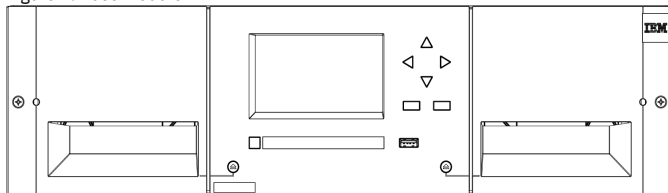


Figure 2. Expansion Module

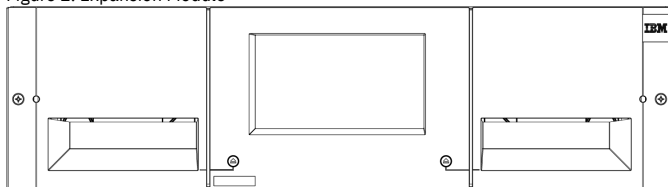
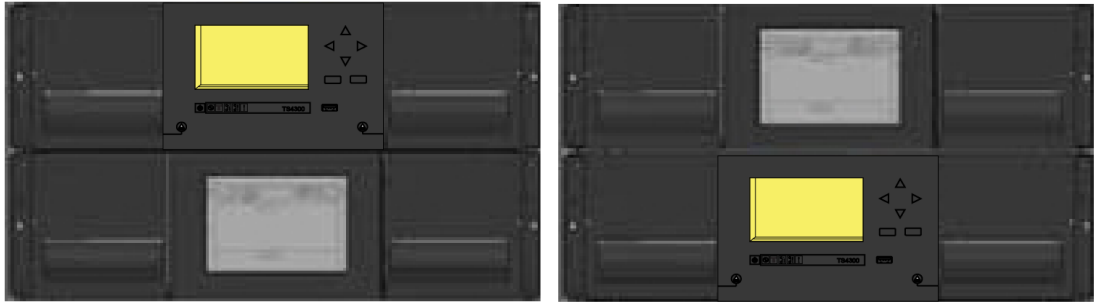
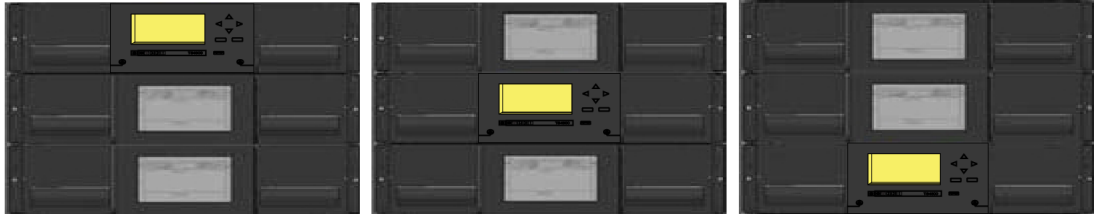


Table 1. Library configurations. Base module displays are yellow.

Module quantity	Supported library configuration
1 module library Base Module only	Figure 3. Base Module

Module quantity	Supported library configuration
2 module library Base Module, and 1 Expansion Module	Figure 4. 2 module library 
3 module library Base Module, and 2 Expansion Modules	Figure 5. 3 module library 
4 module library Base Module, and 3 Expansion Modules	Figure 6. 4 module library

Module quantity	Supported library configuration
	
5 module library Base Module, and 4 Expansion Modules	Figure 7. 5 module library 

Module quantity	Supported library configuration
6 module library Base Module, and 5 Expansion Modules	Figure 8. 6 Module library 
7 module library Base Module, and 6 Expansion Modules	Figure 9. 7 module library 

- [Components](#)

Components

- [Front panel](#)
- [Rear panel](#)
- [Magazines](#)
- [Accessor](#)
- [Power supply](#)

Front panel

Figure 1. Front panel

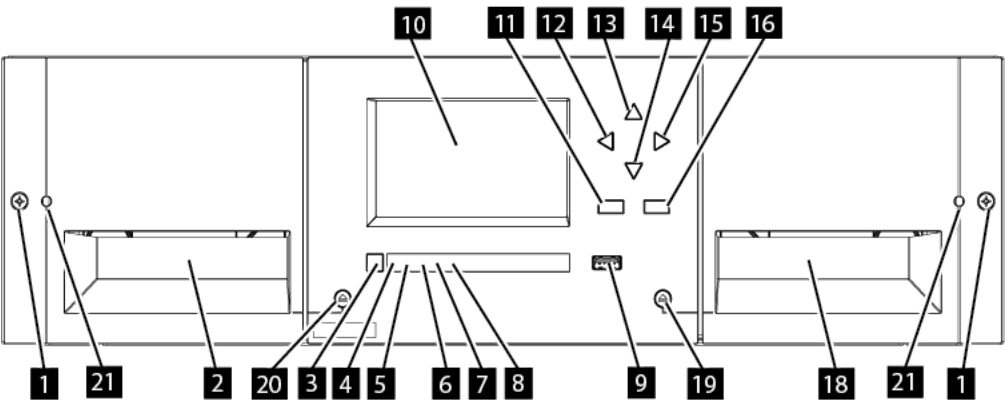


Table 1. Front panel descriptions

Number	Item	Comments
1	Screw holes for attachment to a rack	
2	Left magazine access handle	
3	Power	Base Module only
4	Ready LED, Green	Base Module only
5	Unit Identification LED, Blue	Base Module only
6	Clean LED, Amber	Base Module only
7	Attention LED, Amber	Base Module only
8	Error LED, Amber	Base Module only
9	USB port	Base Module only
10	Operator Panel display	Base Module only
11	Back/Return button	Base Module only
12	Navigation button - Left	Base Module only
13	Navigation button - Up	Base Module only
14	Navigation button - Down	Base Module only
15	Navigation button - Right	Base Module only
16	Enter button	Base Module only
18	I/O station/Right magazine access handle	
19	Right magazine button	
20	Left magazine button	
21	Manual magazine release hole	

Rear panel

Figure 1. Rear panel

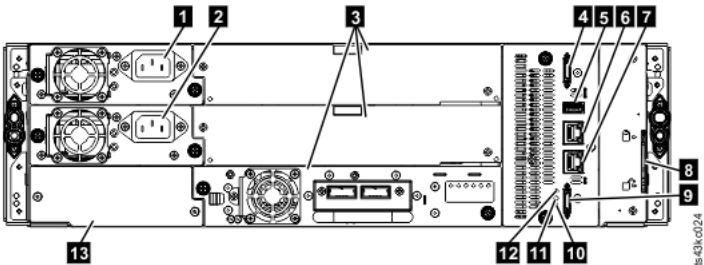


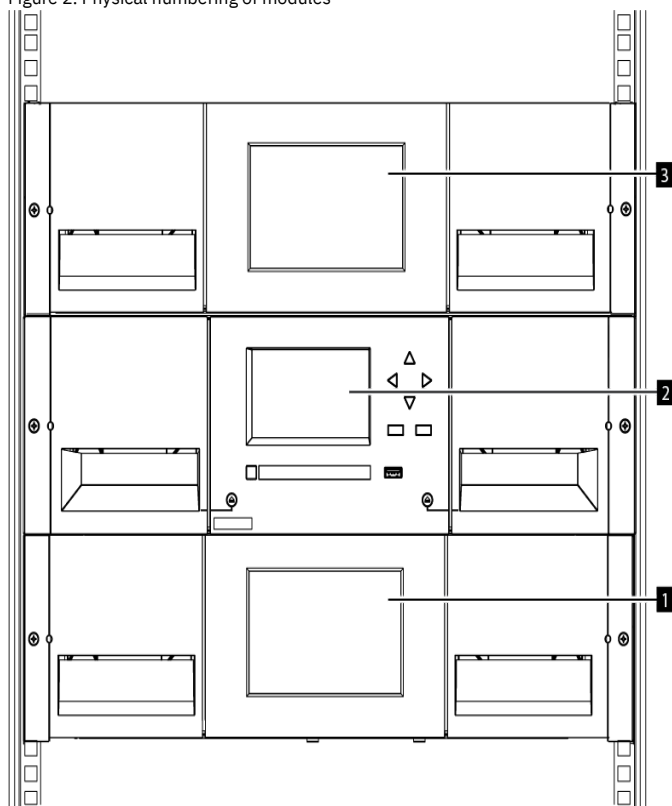
Table 1. Rear panel descriptions

Number	Item	Comments
1	Power supply 1	Standard on Base Module Optional on Expansion Module (required if drives are present)
2	Power supply 2	Optional on Base Module and Expansion Modules
3	Tape drive bays	Full high drives take up 2 bays and can only be installed in the lower two drive bays.
4	Upper Expansion Module connection port	
5	USB port	Base Module only
6	Ethernet port B	Base Module only (secondary port for service usage)
7	Ethernet port A	Base Module only
8	Module alignment mechanism	
9	Lower Expansion Module connection port	
10	Unit Identifier LED, Blue	
11	Controller Error LED, Yellow	
12	Controller Health Status LED, Green	
13	Product Serial Number, Tag location	

Physical and logical addresses of modules

The library assigns each module a unique address to indicate its physical location, shown in [Figure 2](#). The physical numbering is bottom up on all modules.

Figure 2. Physical numbering of modules



Magazines

Each module contains two magazines, holding up to 40 cartridges.

Figure 1. Left magazine

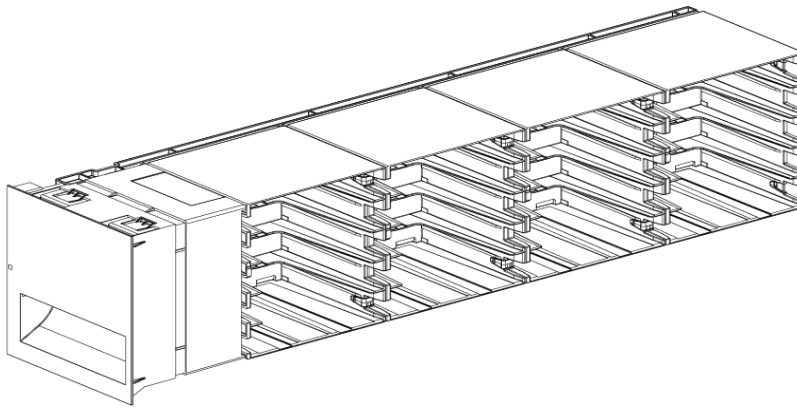
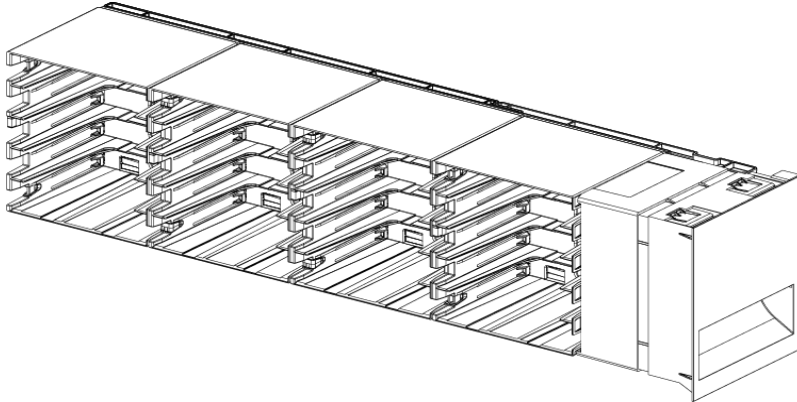


Figure 2. Right magazine



The library assigns each slot in a magazine a unique number to indicate its physical location. This numbering is shown on the Cartridges page of the Management GUI.

Table 1. Physical numbering of storage slots - bottom module

Front of the Left Magazine	5	10	15	20	Drives	25	30	35	40	Front of the Right Magazine
	4	9	14	19		24	29	34	39	
	3	8	13	18		23	28	33	38	
	2	7	12	17		22	27	32	37	
	1	6	11	16		21	26	31	36	

Physical numbering of storage slots starts with the left magazine of the lowest module of your library. As modules are added above this module, the numbering continues, starting with the lower left slot of the left magazine, and adding 40 storage slots for each additional module.

Note: For libraries with serial numbers before 7800K0K, the slots in the lowest row (1, 6, 11, 16, 21, 26, 31, 36) of the bottom module are inaccessible, and can contain a 4-slot I/O station only.

Each module can be configured to contain an I/O Station, or one I/O Station can be accessed by several modules. I/O Stations consist of five slots in the front column of the right magazine of each module, except in older libraries that could not access the bottom row of the lowest module. To enable or disable I/O Stations, go to Library > Modules and Magazines > Actions > Enable or Disable I/O Station in the Management GUI.

Important: Ensure that no tapes are in the slots before the I/O Station is enabled or disabled.

Accessor

The accessor is composed of several components: robotic assembly, spooling mechanism, and bar code reader.

- The robotic assembly has fingers that enable it to grab tape cartridges and move them to and from the I/O station, storage slots, and drives.

- The spooling mechanism has a cable that extracts and retracts based on movement of the robotic assembly. This cable provides communication between the robotic assembly and the library main board.
- The high-speed bar code reader is a part of the Base Module. The bar code reader provides inventory feedback to the host application, Operator Panel display, and Management GUI by reading cartridge bar code labels. The library stores the customized inventory data in memory. Library firmware supports a 6 or 8 character volume serial number (VOLSER) on the bar code label on the tape cartridge. It is highly recommended to use bar code labeled cartridges. See [Bar code labels](#).

Note: For libraries with S/N 7800K0K and higher, review the [Minimum firmware levels for common library features](#).

Power supply

The library provides a single power supply with each library. However, a secondary redundant power supply for the base module can be added. In addition, each expansion module with drives installed requires a power supply. See [Optional features](#).
Remember: For a single module (expansion or base) having two power supplies, the second power supply is deemed as redundant. However, if two or more modules have their individual power supply, then there is no redundant power supply.

Figure 1. Power supply rear panel LEDs

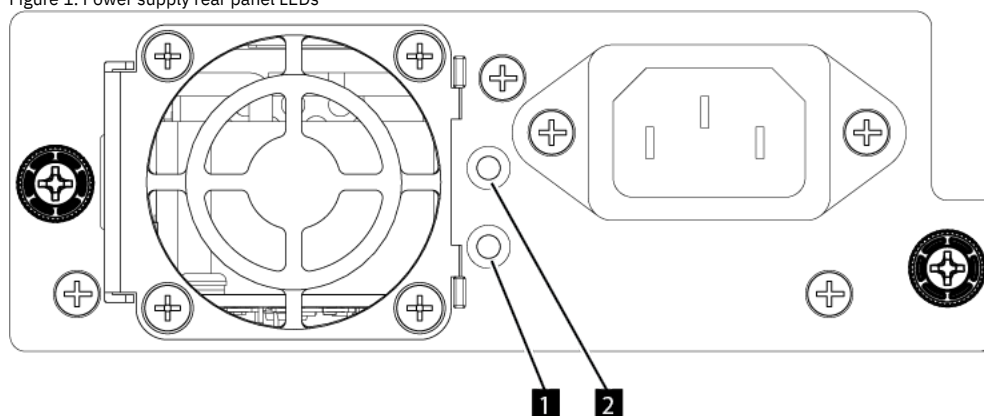


Table 1. Power supply LEDs

Number	Color	Description
1	Green	Module powered ON.
2	White	AC power is connected.

User interfaces

This library has two user interfaces: the Management GUI and the Operator Panel.

- [The Management GUI](#) - With the Management GUI, you can monitor, configure, and control the library from a web browser. The Management GUI hosts a dedicated, protected Internet site that displays a graphical representation of the library. For information on network connectivity, see [Network connectivity](#).
- [The Operator Panel](#) - With the Operator Panel, you can monitor, configure, and control the library from the front panel. Functions are limited to those applicable to being in front of the library.

Supported tape drives

The library supports LTO6 and later tape drives. Each drive requires a SAS or Fibre Channel cable.

For minimum and maximum storage configurations, see [Table 2](#).

Table 1. Drive information and performance specification for full-height drives

Generation	LTO 9	LTO 8	LTO 7	LTO 6
Inquiry	ULT3580-TD9 ULTRIUM-TD9	ULT3580-TD8 ULTRIUM-TD8	ULT3580-TD7 ULTRIUM-TD7	ULT3580-TD6 ULTRIUM-TD6
Interface (speed)	FC (8 Gb) SAS (12 Gb)	FC (8 Gb)	FC (8 Gb)	FC (8 Gb)
Native data rate	400 Mb/s (L9) 360 Mb/s (L8)	360 Mb/s (L8) 300 Mb/s (M8) 300 Mb/s (L7)	300 Mb/s (L7) 160 Mb/s (L6) 140 Mb/s (L5)	160 Mb/s (L6) 140 Mb/s (L5) 120 Mb/s (L4)
Sustained data rate (L6, L7, L8 and L9 compressed at 2.5:1 compression; L5 and earlier compressed at 2:1 compression)	FC 750 Mb/s (L9) 750 Mb/s (L8) SAS 1000 Mb/s (L9) 900 Mb/s (L8)	FC 750 Mb/s (L8) 750 Mb/s (M8) 750 Mb/s (L7)	FC 750 Mb/s (L7) 400 Mb/s (L6) 280 Mb/s (L5)	FC and SAS 400 Mb/s (L6) 280 Mb/s (L5) 240 Mb/s (L4)

Generation	LTO 9	LTO 8	LTO 7	LTO 6
Burst data rate	800 Mb/s FC 1200 Mb/s SAS	800 Mb/s FC	800 Mb/s FC	800 Mb/s FC 600 Mb/s SAS
Nominal load-to-ready time	17 seconds	15 seconds	15 seconds	12 seconds
-Initialized tape	17 seconds	NA	NA	NA
-Uninitialized tape ²	40-120 minutes	NA	NA	NA
Nominal unload time	30 seconds	24 seconds	20 seconds	17 seconds
Average space record time from load point	45 seconds	59 seconds	56 seconds	62 seconds
Average rewind time (REWIND command)	55 seconds	59 seconds	60 seconds	NA
Average rewind time (part of UNLOAD command, dependent on mount activity)³				
Less than 5 Gb of contiguous data transferred	55 seconds	59 seconds	60 seconds	NA
5 Gb to 50 Gb of contiguous data transferred	110 seconds	59 seconds	60 seconds	NA
All other types of mount activity	165 seconds	59 seconds	60 seconds	NA

¹ By using the built-in data-compression capability of the tape drive, greater data rates than the native data transfer rate are achieved. However, the actual throughput is a function of many components, such as the host system processor, disk data rate, block size, data compression ratio, SAS bus capabilities, and system or application software.

² Cartridge initialization time can vary. See [Media optimization](#) for more information.

³ See [Archive mode unthread](#) for more information.

Remember:

- All sustained data rates depend on the capabilities of the interconnect.
- Drive performance varies with media generation and drive interface (SAS/FC).

Table 2. Drive information and performance specification for half-height drives

Generation	LTO 9	LTO 8	LTO 7	LTO 6
Inquiry	ULT3580-HH9 ULTRIUM-HH9	ULT3580-HH8 ULTRIUM-HH8	ULT3580-HH7 ULTRIUM-HH7	ULT3580-HH6 ULTRIUM-HH6
Interface (speed)	FC (8 Gb) SAS (12 Gb)	FC (8 Gb) SAS (6 Gb)	FC (8 Gb) SAS (6 Gb)	FC (8 Gb) SAS (6 Gb)
Native data rate	300 Mb/s (L9) 300 Mb/s (L8)	300 Mb/s (L8) 300 Mb/s (M8) 300 Mb/s (L7)	300 Mb/s (L7) 160 Mb/s (L6) 140 Mb/s (L5)	160 Mb/s (L6) 140 Mb/s (L5) 120 Mb/s (L4)
Sustained data rate (L6, L7, L8 and L9 compressed at 2.5:1 compression; L5 and earlier compressed at 2:1compression)	FC 700 Mb/s (L9) 700 Mb/s (L8) SAS 720 Mb/s (L9) 720 Mb/s (L8)	FC 700 Mb/s (L8) 700 Mb/s (M8) 700 Mb/s (L7) SAS 600 Mb/s (L8) 540 Mb/s (M8) 500 Mb/s (L7)	FC 700 Mb/s (L7) 400 Mb/s (L6) 280 Mb/s (L5) SAS 500 Mb/s (L7) 400 Mb/s (L6) 280 Mb/s (L5)	FC and SAS 400 Mb/s (L6) 280 Mb/s (L5) 240 Mb/s (L4)
Burst data rate	800 Mb/s FC 1200 Mb/s SAS	800 Mb/s FC 600 Mb/s SAS	800 Mb/s FC 600 Mb/s SAS	800 Mb/s FC 600 Mb/s SAS
Nominal load-to-ready time	16 seconds	15 seconds	15 seconds	12 seconds
-Initialized tape	16 seconds	NA	NA	NA
-Uninitialized tape ²	40-132 minutes	NA	NA	NA
Nominal unload time	56 seconds	24 seconds	20 seconds	17 seconds
Average space record time from load point	65 seconds	59 seconds	56 seconds	62 seconds
Average rewind time (REWIND command)	62 seconds	59 seconds	60 seconds	NA
Average rewind time (part of UNLOAD command, dependent on mount activity)³				
Less than 5 Gb of contiguous data transferred	62 seconds	59 seconds	60 seconds	NA
5 Gb to 50 Gb of contiguous data transferred	124 seconds	59 seconds	60 seconds	NA
All other types of mount activity	186 seconds	59 seconds	60 seconds	NA

¹ By using the built-in data-compression capability of the tape drive, greater data rates than the native data transfer rate are achieved. However, the actual throughput is a function of many components, such as the host system processor, disk data rate, block size, data compression ratio, SAS bus capabilities, and system or application software.

² Cartridge initialization time can vary. See [Media optimization](#) for more information.

³ See [Archive mode unthread](#) for more information.

Remember:

- All sustained data rates depend on the capabilities of the interconnect.
- Drive performance varies with media generation and drive interface (SAS/FC).

- **Control path drives**

A control path is a logical path to the library.

- **Mixed drives**

All supported generations of LTO tape drives and cartridges can be in the same physical library and within a single module.

- [Drive sled back panels](#)
Indicator LEDs are included on all drive sled back panels.
- [Physical and logical addresses of drives](#)
The library assigns each tape drive a unique address to indicate its physical and logical location.

Control path drives

A control path is a logical path to the library.

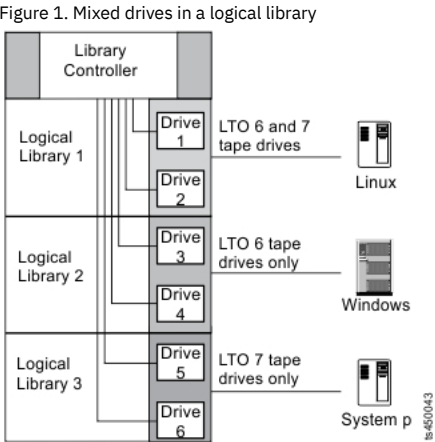
A control path is the path for **SCSI Medium Changer** commands that are sent by a server to control a specific logical library. The library has no direct SCSI connection to a host server. When a software host server communicates with the library, it sends the communication by way of a tape drive. The tape drive is designated as a control path drive.

Mixed drives

All supported generations of LTO tape drives and cartridges can be in the same physical library and within a single module.

This library supports a mixture of LTO drive types in a logical library. Some independent software vendors (ISVs) support mixed drive types within a logical library and other do not. Some ISVs that support mixed drive types might have restrictions. For details, contact your ISV.

[Figure 1](#) shows examples of methods for mixing LTO drive types in a logical library.



Drive sled back panels

Indicator LEDs are included on all drive sled back panels.

Six indicator LEDs are included on all drive sleds as shown in [Figure 1](#).

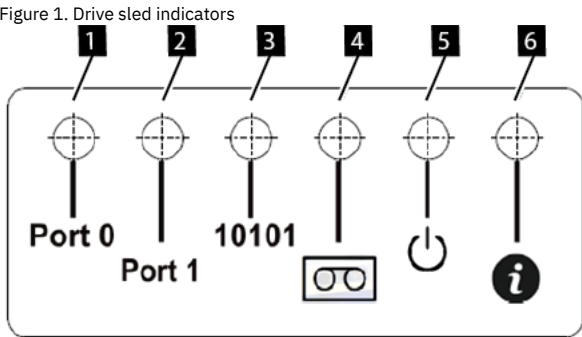


Table 1. Drive sled indicators

Number	Description
1	Port 0 activity
2	Port 1 activity
3	Library communication
4	Cartridge present
5	Power
6	Beacon /UID

Figure 2. Half-height SAS dual port

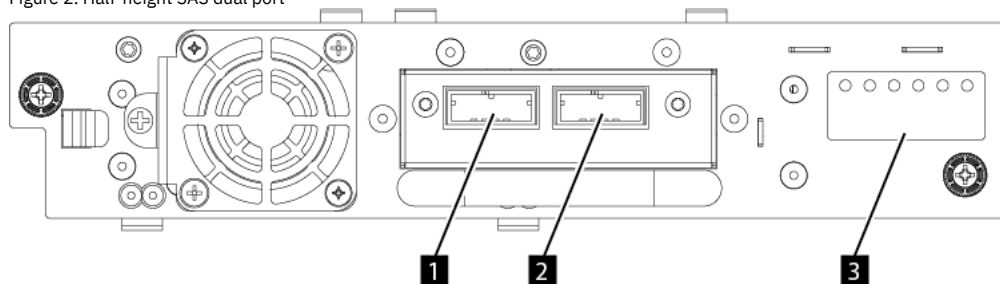


Table 2. Half-height SAS dual port

Number	Description
1	SAS port 0
2	SAS port 1
3	Drive sled indicators (see Figure 1)

Figure 3. Half-height FC single port

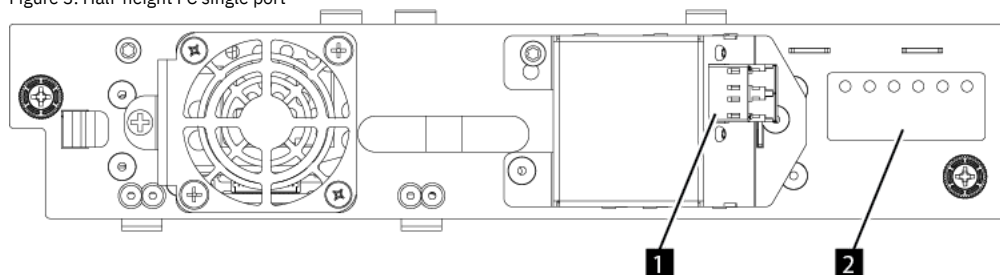


Table 3. Half-height FC single port

Number	Description
1	FC port 0
2	Drive sled indicators (see Figure 1)

Figure 4. Full-height FC dual port

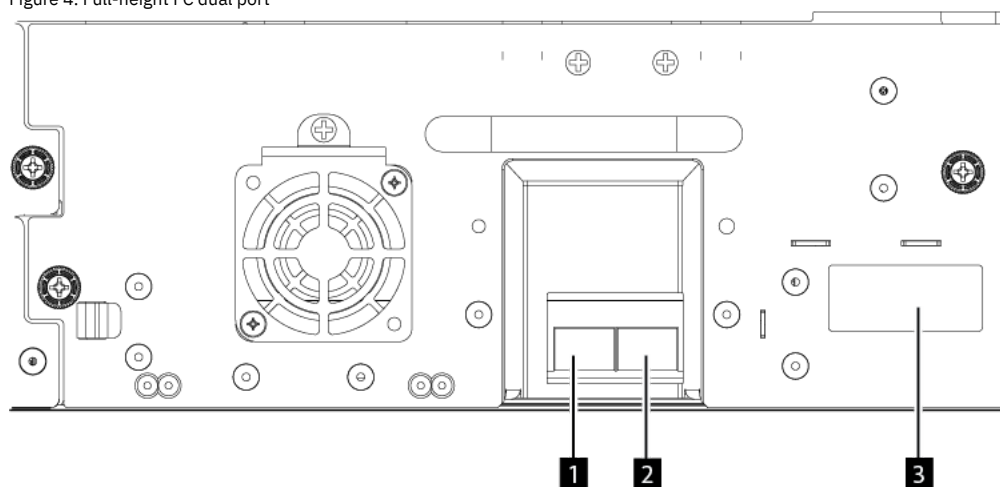


Table 4. Full-height FC dual port

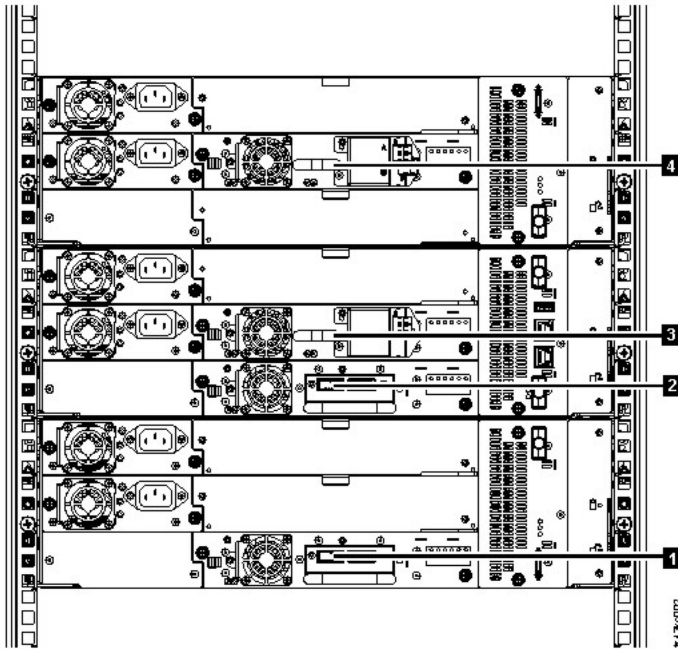
Number	Description
1	FC port 0
2	FC port 1
3	Drive sled indicators (see Figure 1)

Physical and logical addresses of drives

The library assigns each tape drive a unique address to indicate its physical and logical location.

The library assigns each tape drive a unique address to indicate its physical location, which is shown in [Figure 1](#). The physical numbering is bottom up on all drives. This information is shown on the Drives page of the Management GUI.

Figure 1. Physical numbering of drives



The library assigns each tape drive a SCSI element address that consists of a value that defines a logical location in the library to the SCSI interface. It is assigned and used by the application when the host server processes SCSI commands. The SCSI element address for a drive is unique to the location of the drive. It does not vary based on other drives in the library. See [library partitioning](#).

Drive features

- **Media optimization**
Media optimization is a new feature for the LTO9 tape drive with L9/LZ media.
- **Recommended access order (RAO) open function**
RAO enables tape control applications to accelerate the retrieval of a certain number of files from a single tape thereby reducing the seek time between those files.
- **Archive mode unthread**
- **Speed matching**
To improve system performance, the drive uses a technique that is called *speed matching* to dynamically adjust its native (uncompressed) data rate to the slower data rate of a server.
- **Channel calibration**
System performance is optimized by *channel calibration*.
- **Data cartridge capacity scaling**
Capacity scaling enables faster random access to data.
- **Power management**
- **Encryption**
The LTO tape drive supports host Application Managed Encryption (AME), with T10 encryption methods.

Media optimization

Media optimization is a new feature for the LTO9 tape drive with L9/LZ media.

The increased number of tracks used to write data on tape requires greater precision. Media optimization creates a referenced calibration for each cartridge that enables the tape drive's intelligent alignment to optimize data placement. LTO-9 media optimization enhances LTO tape long-term media durability.

It is important to consider when media optimization will be performed:

- Media optimization will be performed on first load of L9/LZ media during initialization.
- Recommendation is to perform first load in the location of deployment, which should be in a stable environment that meets the recommended environmental specification (see [Environment specifications](#) for details).
- Media optimization is a one-time operation that can be completed on any drive in the environment, enabling the media to be used across all tape drives without further optimization.

Other considerations for media optimization:

- Media optimization averages 40 minutes per first load of a cartridge to a tape drive. Although most media optimizations will complete within 60 minutes some media optimizations may take up to 2 hours.
- Interruption of the process is not recommended.
- A different mount will not necessarily improve the time to complete the one-time optimization.

An update to software may be required. Contact your software application provider for more details. Customized software, not provided as a standard market product, may require modification to ensure the software can handle the extended first mount time. For additional details, review the detailed section of the [IBM LTO SCSI Reference](#).

Recommended access order (RAO) open function

RAO enables tape control applications to accelerate the retrieval of a certain number of files from a single tape thereby reducing the seek time between those files.

A feature of the LTO-9 full-height drives is the ability to accept a list of User Data Segments and reorder those User Data Segments into a recommended access order that minimizes the locate portion of the time to read those User Data Segments. This sorted list is called a Recommended Access Order (RAO) list. A User Data Segment (UDS) is defined as a grouping of contiguous logical objects (i.e., logical blocks and filemarks) and is described by partition number, beginning logical object identifier, and ending logical object identifier.

The RAO implementation in LTO produces the best results for performance enhancement when there is little variability in block size or data compression ratio. When the variability in compression ratio or block sizes increase, the accuracy of the locate estimates may be reduced and any potential performance enhancements may be diminished.

Archive mode unthread

The time that is required for an unload depends on how the cartridge has been used during the mount. This is based on the current position and how far from beginning of tape (BOT) the media has been moved since mount. For details on unload performance, see [Supported tape drives](#).

Speed matching

To improve system performance, the drive uses a technique that is called *speed matching* to dynamically adjust its native (uncompressed) data rate to the slower data rate of a server.

With speed matching, the drive operates at different speeds when it is reading or writing the Ultrium 7 or later cartridge format. Native data rates are shown in the table.

Table 1. Performance parameters for full height drives

	Ultrium Generation Media		
	Generation 9 media	Generation 8 media	Generation 7 media
Speed matching data rates (MB/sec)	408	365.0	306.0
	385	341.0	287.52
	366	318.0	268.56
	347	306.4	250.66
	325	273.0	231.86
	305	249.5	213.06
	284	226.0	194.26
	263	203.0	175.46
	244	180.0	157.67
	223	157.5	138.52
	203	135.0	120.11
	177	112.0	101.46

Table 2. Performance parameters for half height drives

	Ultrium Generation Media		
	Generation 9 media	Generation 8 media	Generation 7 media
Speed matching data rates (MB/sec)	284	306.4	306.0
	263	273.0	287.52
	244	249.5	268.56
	223	226.0	250.66
	203	203.0	231.86
	177	180.0	213.06
	-	157.5	194.26
	-	135.0	175.46
	-	112.0	157.67
	-	-	138.52
	-	-	120.11
	-	-	101.46

If the server's net (compressed) data rate is between two of the preceding native data rates, the drive calculates the appropriate data rate at which to operate. Speed matching dramatically reduces backhitch, the condition that occurs when a tape stops, reverses, and restarts motion. A backhitch is usually the result of a mismatch between the data rates of the server and the drive.

Channel calibration

System performance is optimized by *channel calibration*.

System performance is further optimized by a feature that is called *channel calibration*, in which the drive automatically customizes each read/write data channel to compensate for variations in such things as the recording channel's transfer function, the media, and characteristics of the drive head.

Data cartridge capacity scaling

Capacity scaling enables faster random access to data.

The **SET CAPACITY SCSI** command enables a customer to capacity scale a data cartridge to enable faster random access. As an example, a customer can capacity scale a data cartridge to 20% of its normal length that improves the average access time by almost a factor of 5. However, it also reduces the native capacity of the tape to 80 GB160 GB300 GB500 GB1.2 TB.

Power management

The LTO tape drives feature a power management function. This function controls the drive's electronics so that part of the electronics completely turns OFF when circuit functions are not needed for the drive's operation.

Encryption

The LTO tape drive supports host Application Managed Encryption (AME), with T10 encryption methods.

Data encryption is supported by LTO Ultrium 4 and later data cartridges only.

The encryption-enabled drive contains the necessary hardware and firmware to encrypt and decrypt host tape application data. Encryption policy and encryption keys are provided by the host application. A drive digital certificate is installed at manufacturing time. Each drive receives a unique serial number and certificate. The T10 Application can validate each drive instance by checking the drive's digital certificate.

For details, see the *IBM® Tape Device Drivers Encryption Support* documentation, and the *IBM LTO Ultrium Tape Drive SCSI Reference* documentation.

Supported tape cartridges

Tape media is available in the following types:

- Data cartridge
- WORM (Write Once, Read Many) cartridge
- Cleaning cartridge

All generations contain 1/2-inch, dual-coat, magnetic tape.

Table 1. Media drive compatibility

Tape Cartridge	LTO 9 Drive	LTO 8 Drive	LTO 7 Drive	LTO 6 Drive	LTO 5 Drive	LTO 4 Drive	LTO 3 Drive	LTO 2 Drive	LTO 1 Drive
Ultrium 9	Read/write								
Ultrium 8	Read/write	Read/write							
Ultrium M8		Read/write							
Ultrium 7		Read/write	Read/write						
Ultrium 6			Read/write	Read/write					
Ultrium 5			Read only	Read/write	Read/write				
Ultrium 4				Read only	Read/write	Read/write			
Ultrium 3					Read only	Read/write	Read/write		
Ultrium 2						Read only	Read/write	Read/write	
Ultrium 1							Read only	Read/write	Read/write

Table 2. Media Information

LTO Generation	Native Data Capacity	Application Design Capacity	Bar code Label ¹	Load/Unload Cycles	Recording Format ²	Color
Ultrium 9 ³	18 TB (45 TB at 2.5:1 compression)	17.4 TB	xxxxxxL9 WORM: xxxxxxLZ	20,000	Reads and writes data on 8960 tracks, 32 tracks at a time.	Teal WORM: Teal and Silvery gray
Ultrium 8	12 TB (30 TB at 2.5:1 compression)	11.6 TB	xxxxxxL8 WORM: xxxxxxLY	20,000	Reads and writes data on 6656 tracks, 32 tracks at a time.	Burgundy WORM: Burgundy and Silvery gray
Ultrium M8 ⁴	9 TB (22.5 TB at 2.5:1 compression)	8.37 TB	xxxxxxM8	20,000	Reads and writes data on 3584 tracks, 32 tracks at a time.	Purple
Ultrium 7	6 TB (15 TB at 2.5:1 compression)	NA	xxxxxxL7 WORM: xxxxxxLX	20,000	Reads and writes data on 3584 tracks, 32 tracks at a time.	Purple WORM: Purple and Silvery gray
Ultrium 6	2.5 TB (6.25 TB at 2.5:1 compression)	NA	xxxxxxL6 WORM: xxxxxxLW	20,000	Reads and writes data on 2176 tracks, 16 tracks at a time.	Black WORM: Black and Silvery gray
Ultrium 5	1.5 TB (3 TB at 2:1 compression)	NA	xxxxxxL5 WORM: xxxxxxLV	20,000	Reads and writes data on 1280 tracks, 16 tracks at a time.	Burgundy WORM: with Slate Gray

LTO Generation	Native Data Capacity	Application Design Capacity	Bar code Label ¹	Load/Unload Cycles	Recording Format ²	Color
Ultrium 4	800 GB (1.6 TB at 2:1 compression)	NA	xxxxxxL4 WORM: xxxxxxLU	20,000	Reads and writes data on 896 tracks, 16 tracks at a time.	Green WORM: Green and Silvery gray
Ultrium 3	400 GB (800 GB at 2:1 compression)	NA	xxxxxxL3 WORM: xxxxxxLT	10,000	Reads and writes data on 704 tracks, 16 tracks at a time.	Slate Blue WORM: Slate Blue and Silvery gray
Ultrium 2	200 GB (400 GB at 2:1 compression)	NA	xxxxxxL2	10,000	Reads and writes data on 512 tracks, 8 tracks at a time.	Purple
Ultrium 1	100 GB (200 GB at 2:1 compression)	NA	xxxxxxL1	5000	Reads and writes data on 384 tracks, 8 tracks at a time.	Black

¹You can order tape cartridges with the bar code labels included, or you can order custom labels.

²When tape is processed in the cartridges, Ultrium tape drives use a linear, serpentine recording format.

³Cartridge initialization time may vary. For more information, see [Media optimization](#).

⁴LTO type M8 cartridge

For additional information, see LTO media.

Library functions

The library provides many specific functions, such as random or sequential operating mode, encryption, library sharing, path failover, and alerts and logging.

- [Encryption](#)
All supported tape drives in this library support encryption.
- [Library sharing](#)
The library can be configured into one or more logical libraries that can be shared by multiple applications.
- [Control path failover, Data path failover, and load balancing](#)
The path failover feature ensures the use of a redundant communication path when the primary path fails.
- [Alerts and logging](#)
The library sends alerts about the library and attached tape drives, and offers audit-logging to track user actions.
- [Random and Sequential Logical Library modes](#)
A Logical Library can be configured in one of two modes: Random and Sequential.

Encryption

All supported tape drives in this library support encryption.

The encryption enabled drive contains the necessary hardware and firmware to encrypt and decrypt host tape application data. Encryption policy and encryption keys are provided by the host application or host server. A drive digital certificate is installed at manufacturing time. Each drive receives a unique serial number and certificate. The T10 application might validate each drive instance by checking the drive's digital certificate.

The library provides these options.

1. Encryption disabled
2. Application Managed Encryption (AME)
3. Library Managed Encryption (LME). LME is a built-in feature that is enabled by using a purchased license. The LME feature can be ordered from the factory, or you can order it as a field upgrade. To order a feature, contact your IBM Sales Representative or Business Partner. See [Optional Features](#). For configuration details, see [Configuring Library Managed Encryption](#).

The default is Application Managed Encryption.

Note: All encryption settings must be configured or reverified in the drive after any library or drive reset. A new drive might be added or an existing drive might be swapped with another drive.

Library sharing

The library can be configured into one or more logical libraries that can be shared by multiple applications.

It is advantageous to be able to share a single physical library between heterogeneous or homogeneous applications. However, some applications (and some servers) do not allow for sharing a library between systems.

The library Management GUI provides two methods for logical library configuration.

1. A quick configuration for a simple one logical library configuration
2. An advanced configuration action for a multiple logical library configuration

Note: When any number of drives are loaded, a warning message appears when the Manage Logical Library (Expert Mode) wizard is accessed.

The second method gives the ability to create configurations that enable the library to process commands from multiple heterogeneous applications (such as an IBM® System p application and a Windows application) and multiple homogeneous applications (for example, the same application run by several System p servers). See [Advanced library configuration](#).

Control path failover, Data path failover, and load balancing

The path failover feature ensures the use of a redundant communication path when the primary path fails.

Command failures and time outs are costly. You want your library to run smoothly and efficiently. Path failover capabilities allow the IBM® device driver to resend a command to an alternate path. The alternate path can include another host bus adapter (HBA), Storage Area Network (SAN), or library control path drive. The device driver initiates error recovery and continues the operation on the alternate path without interrupting the application.

Path failover and load balancing are built-in features that are enabled by using a purchased license. The path failover feature can be ordered from the factory, or you can order it as a field upgrade. The path failover feature (FC 1682) is activated on the Management GUI. To order features, contact your IBM® Sales Representative or Business Partner. See [Optional features](#).

Two types of path failover capabilities exist: *control path failover (CPF)* and *data path failover (DPF)*. *Control* refers to the command set that controls the library (the **SCSI Medium Changer** command set on LUN 1 of the tape drives). *Data* refers to the command set that carries the customer data to and from the tape drives (the SCSI-3 Stream Commands (SSC) device on LUN 0 of the tape drives). Path failover means the same thing in both. Path failover is where redundancy is in the path from the application to the intended target (the library accessor or the drive mechanism), the device driver transparently fails over to another path in response to a break in the active path.

Both types of failover include host-side failover when configured with multiple HBA ports into a switch. But CPF includes target-side failover through the control paths that are enabled on more than one tape drive. DPF includes target-side failover for the dual-ported tape drives that are supported by the library.

DPF includes load balancing of the HBAs because the channel is a data-intensive path (the control path carries little data, so load balancing is not an issue). The dynamic load balancing support optimizes resources for devices that have physical connections to multiple HBAs in the same machine. When an application opens a device where multiple HBA paths are 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 updates the usage on the HBA assigned to the application when the device is closed. Dynamic load balancing uses all HBAs whenever possible and balances the load between them to optimize the resources in the machine.

Both CPF and DPF require the use of the IBM device driver. They are supported exclusively with products that bear the IBM logo on the operating systems that is indicated in [Table 1](#).

[Table 1](#) summarizes the differences between CPF, DPF, and load balancing.

Table 1. Differences between CPF and DPF

Characteristic	CPF	DPF and Load Balancing
Device type	SMC ¹	SSC ²
LUN ³	LUN 1	LUN 0
Host-side failover	Yes	Yes ⁶
Target-side failover	Yes	Yes ⁶
IBM Device driver required	Yes	Yes
Supported operating systems ⁴	AIX®, SuSE Linux, Red Hat Enterprise Linux, Solaris, Windows	AIX, SuSE Linux, Red Hat Enterprise Linux, Solaris, Windows ⁵ (DPF only)
Order feature to obtain license	Yes	Yes
Notes: 1. SMC = SCSI-3 Medium Changer Specification (library) 2. SSC = SCSI-3 Stream Commands (drive) 3. LUN = logical unit number 4. See Host connectivity for details. 5. Load balancing is not supported on Windows 6. Full-height tape drives only		

For information about using these features, see the *IBM Tape Device Drivers Installation and User's Guide* (GA32-0565).

Alerts and logging

The library sends alerts about the library and attached tape drives, and offers audit-logging to track user actions.

- **TapeAlert Support:** The tape library is compatible with TapeAlert technology, which provides error and diagnostic information about the drives and the library to the host application. The library provides this error and diagnostic information as TapeAlert flags that are reported to the application by the **SCSI LOG SENSE** command. See [TapeAlert flags](#).
- **Email (SMTP - Simple Mail Transfer Protocol) Notifications:** The library can configure email notification of library events. The library must have network access to an SMTP server. See [Locating Management functions](#).
- **Remote Logging (rsyslog):** The library can send syslog (system log) notifications to a configured remote (rsyslog) server. When system events occur, the TS4300 tape library creates a log of these events. With this notification feature configured, the library sends a notification of the event to the syslog server. The syslog server keeps its own log of system events. (The syslog server is a customer-provided server.) See [Locating Management functions](#).
- **SNMP Support:** The Simple Network Management Protocol (SNMP) allows the library to send alerts over a LAN network to a monitoring server. Occasionally, the library might encounter situations that you want to know about. These situations can be conditions that affect the library performance, such as an open door that causes the library to stop. You might also want to log user actions, such as a cartridge move or export that is initiated from the Management GUI. SNMP messages can alert you of these conditions.

The library provides a standard TCP/IP protocol that is called SNMP to send alerts about conditions over a TCP/IP LAN network to an SNMP monitoring server. These alerts are called SNMP traps. Using the information that is supplied in each SNMP trap, the monitoring server (together with customer-supplied software) can alert operations staff of possible problems or operator interventions that occur. Many monitoring servers can be used to send email or pager notifications when they receive an SNMP alert. See the manual for your network management application.

The monitoring server must be loaded with systems management software that can receive and process the trap. SNMP supports a get and get-response mechanism for an operator to gather more information about a problem or query the library about its status. Through a monitoring server, the operator enters a "get" using SNMP to request information about the library. A get-response is the information that is provided in response to the get. This type of support generally requires an up-to-date library Management Information Base (MIB). The SNMP server's MIB contains units of information that specifically describe an aspect of a system, such as the system name, hardware number, or communications configuration.

IBM provides the MIBs that are supported by the library. They include

- IBM TS4300 MIB
 - Type of information:
 - The status of each drive in the library
 - A list of all cartridges in the library
 - The last trap or TapeAlert message that was sent by the library.
- IBM AUTOMATION QUERY MIB
 - Type of information:
 - Library configuration

The library MIBs can be obtained through the SNMP Notifications Settings page on the Management GUI. See [Locating Management functions](#). An operator cannot change library settings by using SNMP. Settings are changed by using the Management GUI.

SNMP Notification Levels

SNMP provides various levels of notification about specific library events and user actions.

- Inactive – No events are sent.
- Critical – Only critical events are sent.
- + Warnings – Only critical and warning events are sent.
- + Configuration – Only critical, warning, and configuration events are sent.
- + Information – All events are sent.

Simple Network Management Protocol (SNMP) audit logging provides logging information about specific tape library user actions. To configure SNMP, see [Locating Management functions](#).

Random and Sequential Logical Library modes

A Logical Library can be configured in one of two modes: Random and Sequential.

Random Mode

Random Mode is intended to be used by host applications that support SCSI media changer devices. Random Mode is the default.

In Random mode,

- The host application chooses the cartridges that are moved to the drive.
- I/O slots provide the flexibility for the user to add and remove cartridges and the host application is automatically notified of these changes.
- Multiple drives can be assigned to provide parallel processing data operations and redundancy in case of failure.

Sequential Mode

Sequential Mode is intended to be used by host applications that aren't supporting SCSI media changer devices but need to get another cartridge loaded if the current cartridge is full.

In Sequential Mode,

- The library predefines the sequential order that the cartridges are moved to the drive.
- I/O slots are hidden as they aren't assignable to a logical library with sequential mode enabled.
- Only one drive can be assigned to a logical library with sequential mode enabled.
- There's no control path drive and no media changer device is configured to the host server.

Options to consider when Sequential Mode is chosen:

- Basic Function - To initiate use of cartridges, the user issues a **Move Cartridge** command to the drive through the Management GUI. After the load, the host application can begin data I/O activity. When the host application unloads the drive, the library moves the next cartridge into the drive. This behavior is implicit, unless otherwise defined by selection of another option.
- Loop Option - If a move sequence ends because no more cartridges are available in the current logical library, the sequence starts again by loading the first cartridge of the logical library. This option can be chosen with or without the Autoload function.
- Autoload Option - If enabled, the library loads the first cartridge of the logical library to the Sequential Mode tape drive during library startup after inventory scan is finished. This option changes the implicit behavior of the Basic function. This option can be chosen with or without the Loop option.

If storage slots are configured to I/O slots after assignment to a Sequential Mode logical library, they are still considered valid available slots and are used for movements. This option stays until you run the expert wizard again, and then these I/O slots no longer appear in the list of available slots. Finishing the expert wizard then sets the new slot assignment and these I/O slots can no longer be used for movements.

To enable sequential mode, click the Enable Sequential Mode check box in the Basic Logical Library Wizard or Expert Logical Library Wizard.

Planning

The library requires an environment able to accommodate the appropriate space, power, location, and other technical specifications. Use this section as a reference for onsite requirements to allow for optimum operation of the library.

Save your settings in the [Library Configuration Forms](#).

- [Acclimation](#)
Server and storage equipment (racks and frames) must be gradually acclimated to the surrounding environment to prevent condensation.
- [Library Layout and Location requirements](#)
Information for planning the installation and layout of your library, including various specifications for optimal performance.
- [Power cords](#)
Electrical and safety information, and feature codes for purchasing power cords.
- [Network requirements](#)
The library supports an independent customer network.
- [Host requirements](#)
The drive is supported by a wide variety of servers, operating systems, and adapters. Many ways to determine the servers and software that supports this drive are available.
- [Optional features](#)
More features that are available for your library.

Acclimation

Server and storage equipment (racks and frames) must be gradually acclimated to the surrounding environment to prevent condensation.

When server and storage equipment (racks and frames) is shipped in a climate where the outside temperature is below the dew point of the destination (indoor location), there's a possibility that water condensation can form on the cooler inside and outside surfaces of the equipment when the equipment is brought indoors.

Sufficient time must be allowed for the shipped equipment to gradually reach thermal equilibrium with the indoor environment before you remove the shipping bag and energize the equipment. Follow these guidelines to properly acclimate your equipment:

- Leave the system in the shipping bag. If the installation or staging environment allows it, leave the product in the full package to minimize condensation on or within the equipment.
- Allow the packaged product to acclimate for 24 hours.¹ If there are visible signs of condensation (either external or internal to the product) after 24 hours, acclimate the system without the shipping bag for an extra 12 - 24 hours or until no visible condensation remains.
- Acclimate the product away from perforated tiles or other direct sources of forced air convection to minimize excessive condensation on or within the equipment.

¹ Unless otherwise stated by product-specific installation instructions.

Note: Condensation is a normal occurrence, especially when you ship equipment in cold-weather climates. All IBM® products are tested and verified to withstand condensation that is produced under these circumstances. When sufficient time is provided to allow the hardware to gradually acclimate to the indoor environment, there should be no issues with long-term reliability of the product.

Library Layout and Location requirements

Information for planning the installation and layout of your library, including various specifications for optimal performance.

For tabletop installation - Tabletop installations (one Base Module) require no additional hardware.

For rackmount installation - If possible, install the Base Module in the middle of the rack to provide space for the allowed three Expansion Modules above it and three Expansion Modules below it. See [Structure and supported library configurations](#) for details.

Security

The equipment must be located so that access to the equipment can be controlled and monitored. Consider all of these recommended security measures when you're determining where to locate your tape library.

Library location

You're responsible for the security of this library, the cartridges that are contained within the library, and shelf-resident cartridges. To prevent unauthorized access to data, IBM recommends locating the library and all shelf-resident cartridges in an area where access is controlled.

Onsite security measures

You're also responsible for evaluating, selecting, and implementing security features, administrative procedures, and appropriate controls in application systems and communication facilities.

Data security

Data security is accomplished through the Management GUI. See [Locating Management functions](#).

Location requirements

Choose a location that meets the criteria in [Table 1](#).

Table 1. Location requirements

Criteria	Definition
Rack requirements	Standard 19-inch rack (minimum depth of 1 meter) with an appropriate # of Us (Rack Units) of clearance for the planned module quantity (See Table 2 for details.) Minimum Rack Post Spacing: 688.34 mm (27.1 In.) Maximum Rack Post Spacing: 904.24 mm (35.6 In.)
Rack space requirements	3U for the Base Module and 3U for each Expansion Module

Criteria	Definition
Power source	- AC Power Voltage: 100 - 240 VAC - Line Frequency: 50 - 60 Hz - Library is located near AC outlet. The AC power cord must be always easily accessible.
Air quality	- Place the library in an area with minimal sources of particulate contamination. - Avoid areas near frequently used doors and walkways, stacks of supplies that collect dust, printers, and smoke-filled rooms. - Excessive dust and debris can damage tapes and tape drive.

Technical specifications for this library can be referenced in the following tables.

Physical specifications

Table 2. Physical specifications

Characteristic	Product alone	Packaged
Height	133 mm (5.23 in)	330 mm
Width	480 mm (18.89 in) ¹	635 mm
Depth	880 mm (34.6 in) ²	1168 mm
Weight	Base module: 20 Kg Expansion module: 13 Kg	Base module: 25 Kg Expansion module: 19 Kg

¹Includes front covering of rack rails, allowing for magazine opening clearance.
²From the front of the bezel to the back of the fan on an inserted drive sled.

Figure 1. Front bezel height

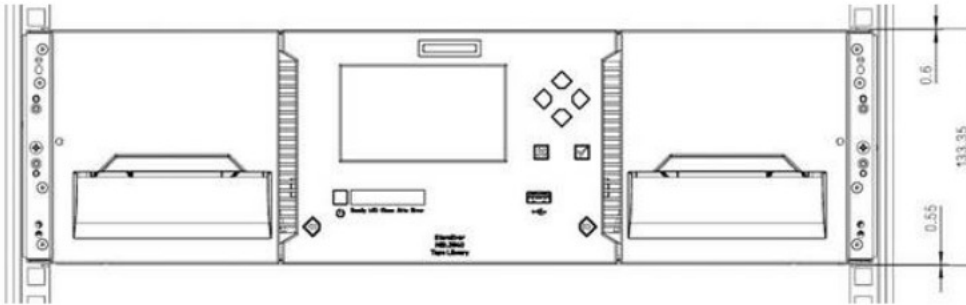
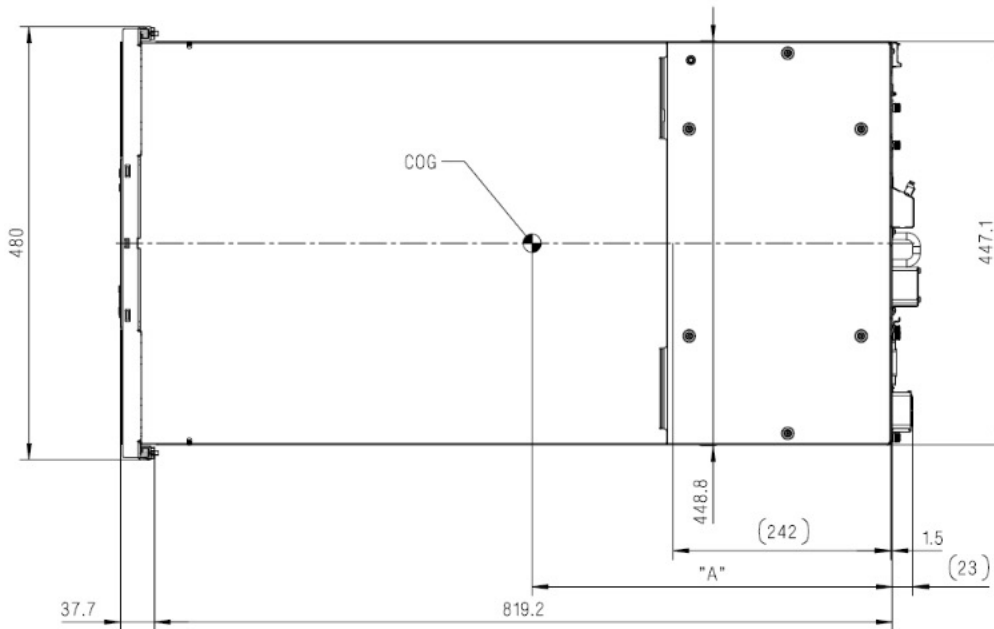


Figure 2. Depth from front of the bezel to back of the fan on an inserted drive sled



Electrical specifications for one module (base or expansion)

Table 3. Electrical specifications for one module (base or expansion)

Characteristic	Specification
Maximum current	3.7 A

Characteristic	Specification
Voltage	100 - 240 V 50/60 Hz
Maximum power consumption	370 W

Equipment environmental specifications

Table 4. Equipment environment specifications for the tape library

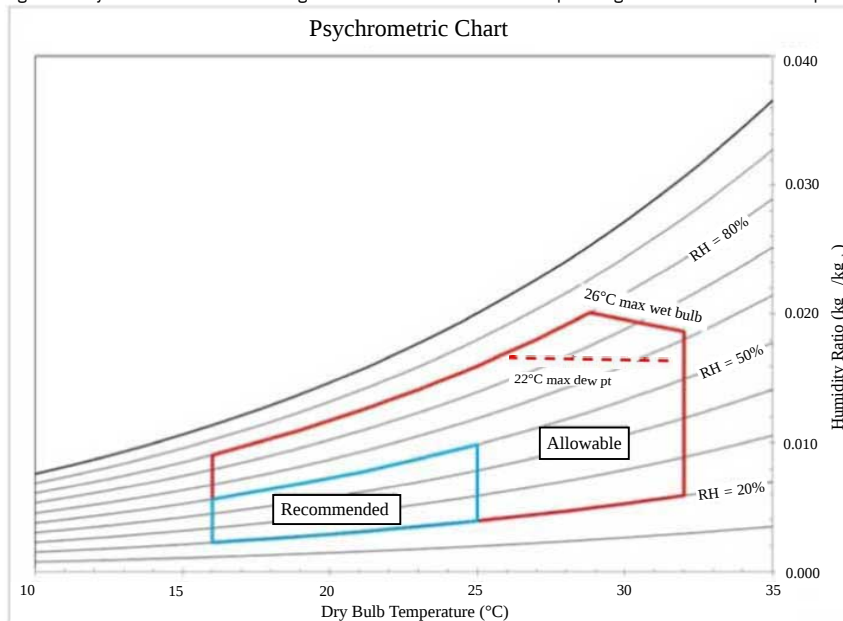
Product operation (equipment is powered on)									
Dry-bulb temperature			Humidity range, non-condensing			Maximum wet-bulb temperature ⁴	Maximum dew point temperature ⁵	Maximum elevation	Dry-bulb temperature
Allowable ²	Recommended ³	Maximum rate of change	Allowable	Recommended	Maximum rate of change				
16°C to 32°C (60°F to 90°F)	16°C to 25°C (60°F to 77°F)	5°C/hour (9°F/hour)	20 to 80% RH	20 to 50% RH	5% RH/hour with no condensation	26°C (79°F)	22°C (72°F)	3050 m (10,000 feet)	5°C to (40°F)

Notes:

- Product equipment is removed from the original shipping container and installed but not in use - for example, during repair, maintenance, or upgrade.
- Derate maximum dry-bulb temperature 1°C/300 m above 900 m (1.8°F/1,000 feet above 3,000 feet).
- Derate maximum recommended dry-bulb temperature 1°C/300 m above 1,800 m (1.8°F/1,000 feet above 6,000 feet).
- Applies to LTO drive generations 1 through 8.
- Applies to LTO 9 drives

Psychrometric chart

Figure 3. Psychrometric chart showing recommended and allowable operating environments for the tape library



Notes:

- The chart is shown in SI (metric) units and a barometric pressure of 101.325 kPa (sea level).
- The recommended operating environment specifies a long-term operating environment that can result in the greatest reliability and energy efficiency.
- The allowable operating environment represents where the equipment has been tested to verify functions.

Gas and particulate exposure

Table 5. Gas and particulate exposure

Contamination	Requirement
Gaseous contamination	Severity level G1 as per ANSI/ISA 71.04-1985 ¹ , which states that the reactivity rate of copper coupons shall be fewer than 300 Angstroms per month ($\text{\AA}/\text{month}$, $\approx 0.0039 \mu\text{g}/\text{cm}^2 \cdot \text{hour}$ weight gain) ² . In addition, the reactivity rate of silver coupons shall be less than 300 $\text{\AA}/\text{month}$ ($\approx 0.0035 \mu\text{g}/\text{cm}^2 \cdot \text{hour}$ weight gain) ³ . The reactive monitoring of gaseous corrosivity should be conducted approximately 5 cm (2 in.) in front of the rack on the air inlet side at one-quarter and three-quarter frame height off the floor or where the air velocity is much higher.
Particulate contamination	Data centers must meet the cleanliness level of ISO 14644-1 class 8. For data centers without airside economizer, the ISO 14644-1 class 8 cleanliness might be met by the choice of the following filtration: - The room air might be continuously filtered with MERV 8 filters. - Air entering a data center might be filtered with MERV 11 or preferably MERV 13 filters. For data centers with airside economizers, the choice of filters to achieve ISO class 8 cleanliness depends on the specific conditions present at that data center. The deliquescent relative humidity of the particulate contamination should be more than 60% RH. ³⁴ . Data centers must be free of zinc whiskers ⁵ .

Contamination	Requirement
Notes: 1. ANSI/ISA-S71.04, 1985. <i>Environmental conditions for process measurement and control systems: Airborne contaminants</i>, Instrument Society of America, Research Triangle Park, NC, 1985. 2. The derivation of the equivalence between the rate of copper corrosion product thickness growth in Å/month and the rate of weight gain assumes that Cu₂S and Cu₂O grow in equal proportions. 3. The derivation of the equivalence between the rate of silver corrosion product thickness growth in Å/month and the rate of weight gain assumes that Ag₂S is the only corrosion product. 4. The deliquescent relative humidity of particulate contamination is the relative humidity at which the dust absorbs enough water to become wet and promote ionic conduction. 5. Surface debris is randomly collected from 10 areas of the data center on a 1.5-cm diameter disk of sticky electrically conductive tape on a metal stub. If examination of the sticky tape in a scanning electron microscope reveals no zinc whiskers, the data center is considered free of zinc whiskers.	

Acoustical specifications

Table 6. Acoustical specifications

Parameter	Measurement
Idling acoustical noise sound power level LwAD in Bels (1 Bel = 10 dB)	6.6
Maximum acoustical noise sound power level LwAD in Bels (1 Bel = 10 dB)	6.8

Power cords

Electrical and safety information, and feature codes for purchasing power cords.

	To avoid electrical shock, a power cord with a grounded attachment plug is provided. Use only properly grounded outlets.		
	Table 1 lists the power cord part number, feature code, the country, or region where the power cord is used, and the plug's standard reference. The last column in the table contains an index number that you can match to a specific receptacle type in Figure 1 .		
	All power cords use an appliance coupler that complies with the International Electrotechnical Commission (IEC) Standard 320, Sheet C13.		
	If the power cord that you receive does not match your receptacle, contact your local dealer.		
	Power cords that are used in the United States and Canada are listed by Underwriter's Laboratories (UL), are certified by the Canadian Standards Association (CSA), and comply with the plug standards of the National Electrical Manufacturers Association (NEMA). For other worldwide geographies, plug standards are listed in Table 1 .		

Table 1. Power cords

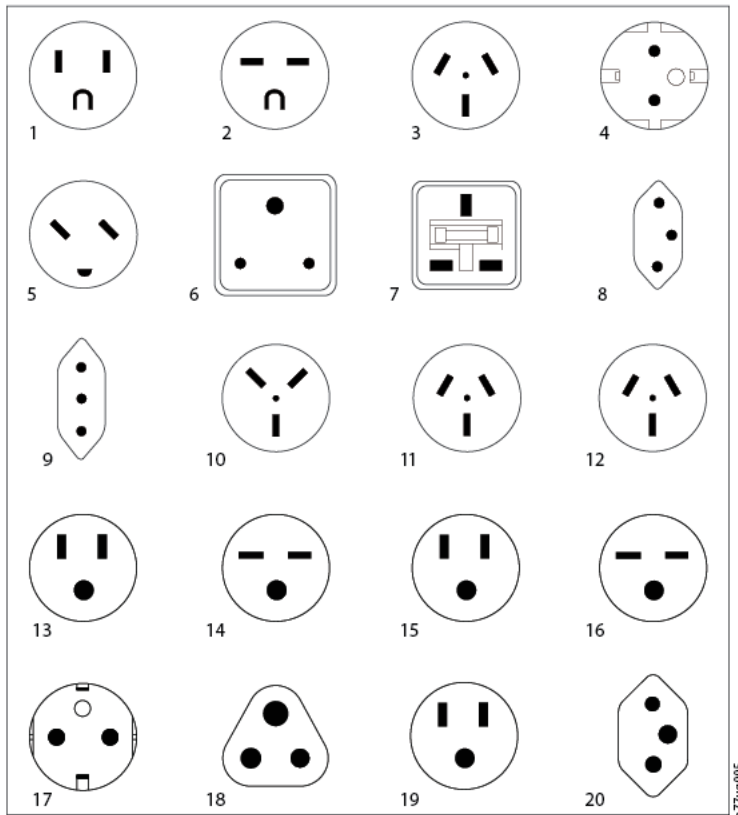
Description, Feature Code (FC), and Part Number (P/N)	Plug Standard Reference	Country or Region	Index Number in Figure 1
US/Canada • 2.8 m, 125 V • FC 9800 • P/N 95P2344	NEMA 5-15P	Aruba, Bahamas, Barbados, Bermuda, Bolivia, Brazil, Canada, Cayman Islands, Colombia, Costa Rica, Curacao, Dominican Republic, Ecuador, El Salvador, Guatemala, Guyana, Haiti, Honduras, Jamaica, Japan, Liberia, Mexico, Netherlands Antilles, Nicaragua, Panama, Peru, Philippines, Saudi Arabia, South Korea, Suriname, Taiwan, Trinidad Tobago, Venezuela, US	1
Chicago • 1.8 m, 125 V • FC 9986 • P/N 39M5080	NEMA 5-15P	Chicago, U.S.A.	1
US/Canada • 2.8 m, 250 V • FC 9833 • P/N 95P2353	NEMA 6-15P	Aruba, Bahamas, Barbados, Bermuda, Bolivia, Brazil, Canada, Cayman Islands, Costa Rica, Curacao, Dominican Republic, Ecuador, El Salvador, Guatemala, Guyana, Haiti, Honduras, Jamaica, Japan, Liberia, Netherlands Antilles, Nicaragua, Panama, Peru, Philippines, Suriname, Taiwan, Thailand, Trinidad Tobago, Venezuela, US	2
Australia • 2.8 m, 250 V • FC 9831 • P/N 95P2352	AS 3112 NZS 198	Australia, China, Colombia, New Zealand, Papua New Guinea, Paraguay, Uruguay, Western Samoa	3

Description, Feature Code (FC), and Part Number (P/N)	Plug Standard Reference	Country or Region	Index Number in Figure 1
France, Germany • 2.8 m, 250 V • FC 9820 • P/N 95P2345	CEE 7 - VII	Afghanistan, Algeria, Andorra, Angola, Aruba, Austria, Belgium, Benin, Brazil, Bulgaria, Burkina Faso, Burundi, Cameroon, Central African Republic, Chad, Congo-Brazzaville, Curacao, Czech Republic, Democratic Republic of Congo, Denmark, Egypt, Finland, France, French Guiana, Germany, Greece, Guinea, Hungary, Iceland, Indonesia, Iran, Ivory Coast, Jordan, Kenya, Korea, Lebanon, Luxembourg, Macau, Malagasy, Mali, Martinique, Mauritania, Mauritius, Monaco, Morocco, Mozambique, Netherlands, Netherlands Antilles, New Caledonia, Niger, Norway, Poland, Portugal, Romania, Russia, Saudi Arabia, Senegal, Spain, Sweden, Sudan, Syria, Togo, Tunisia, Turkey, Yugoslavia, Zaire, Zimbabwe, Vietnam	4
Denmark • 2.8 m, 250 V • FC 9821 • P/N 95P2346	DK2-5A	Denmark	5
South Africa • 2.8 m, 250 V • FC 9829 • P/N 95P2350	SABS 164	Bangladesh, Burma, Pakistan, South Africa, Sri Lanka	6
United Kingdom • 2.8 m, 250 V • FC 9825 • P/N 95P2347	BS 1363	Antigua, Bahrain, Bermuda, Brunei, Channel Islands, China (Hong Kong S.A.R.), Cyprus, Fiji, Ghana, Guyana, India, Iraq, Ireland, Jordan, Kenya, Kuwait, Malaysia, Malawi, Malta, Nepal, Nigeria, Oman, Polynesia, Qatar, Sierra Leone, Singapore, Tanzania, Uganda, UK, United Arab Emirate (Dubai), Yemen, Zambia	7
Switzerland • 2.8 m, 250 V • FC 9828 • P/N 95P2349	SEV S/N 416534	Liechtenstein, Switzerland	8
Italy • 2.8 m, 250 V • FC 9830 • P/N 95P2351	CEI 23- 16	Chile, Ethiopia, Italy, Libya, Somalia	9
Israel • 2.8 m, 250 V • FC 9827 • P/N 95P2348	S11-32-1971	Israel	10
Argentina • 2.8 m, 250 V • FC 9834 • P/N 95P2354	IEC 83-A5	Argentina, Brazil, Colombia, Paraguay, Trinidad Tobago, Uruguay	11
China • 2.8 m, 250 V • FC 9840 • P/N 95P2355	CCEE	People's Republic of China	12

Description, Feature Code (FC), and Part Number (P/N)	Plug Standard Reference	Country or Region	Index Number in Figure 1
Taiwan LV* • 2.8 m, 125 V • FC 9835 • P/N 23R3263	CNS 10917-3	Taiwan	13
Taiwan HV** • 2.8 m, 250 V • FC 9841 • P/N 23R6120	CNS 10917-3	Taiwan	14
Japan LV* • 2.8 m, 125 V • FC 9842 • P/N 23R6121	JIS C8303, C8306	Japan	15
Japan HV** • 2.8 m, 250 V • FC 9843 • P/N 39M5186	JIS C8303, C8306	Japan	16
Korea HV** • 2.8 m, 250 V • FC 9844 • P/N 23R6123	KS C8305, K60884-1	Korea	17
India HV** • 2.8 m, 250 V • FC 9845 • P/N 23R6124	IS 6538	India	18
Brazil LV* • 2.8 m, 125 V • FC 9846 • P/N 39M5233	InMetro NBR 6147	Brazil	19
Brazil HV** • 2.8 m, 250 V • FC 9847 • P/N 23R6126	InMetro NBR 14136	Brazil	20
Rack PDU • FC 9848 • P/N 23R6328			
* Low Voltage ** High Voltage			

[Figure 1](#) shows the plugs that are used by the power cords in [Table 1](#). Match the index number that is beside each plug to the index number in the table.

Figure 1. Types of receptacles



a77ug005

Network requirements

The library supports an independent customer network.

It is the customer's responsibility to provide the proper length Ethernet cable for this connectivity.

The base module controller card has two Ethernet ports, which offer primary and redundant customer network connectivity. See [Rear panel](#).

These connections allow remote viewing and management of the library with the Management GUI.

Note: Have your network settings handy to use for entering on the Operator Panel. Your network settings can also be stored as hardcopy on [Library Configuration Forms](#).

The secondary Ethernet port might be used for service. Three models are available for connection:

- No Ethernet port - Service personnel can connect a laptop to the customer network to use the Management GUI.
- Dedicated secondary Ethernet port - The secondary network port that is dedicated only for service personnel to connect a laptop directly to the library.
- Secondary Ethernet port - The secondary network port can be disconnected and service personnel can use it to connect a laptop directly to the library.

IP range selection

For internal communication between modules, the tape library uses an Ethernet connection with an internal IP address range. To prevent any conflict between the internal IP address range and the external IP addresses, you must select the internal IP range. Choosing the Internal IP address range, and also entering the external IP address information is part of the initial setup of the library.

Supported browsers

IBM supports higher versions of the browsers if the vendors don't remove or disable functions that the product relies upon. For browser levels higher than the versions that are certified with the product, customer support accepts usage-related and defect-related service requests. As with operating system and virtualization environments, if IBM support can't re-create the issue in the lab, the client might be asked to re-create the problem on a certified browser version to determine whether a product defect exists. Defects aren't accepted for cosmetic differences between browsers or browser versions that don't affect the functional behavior of the product. If a problem is identified in the product, defects are accepted. If a problem is identified with the browser, IBM might investigate potential solutions or workarounds that the client can implement until a permanent solution becomes available.

Supported interfaces

This tape library supports the Gigabit Ethernet interface in either auto negotiation or fixed modes of 10 Mbps, 100 Mbps, and 1 Gbps by using half or full duplex. The library supports the following TCP/IP protocols:

IPv4 and IPv6 support

This tape library supports Internet Protocol (IP) addresses in both IPv4 and IPv6 format. Both the integrated management console (IMC) and the management GUI allow the definition of IPv4 and IPv6 addresses. The key proxy determines the IP version that is used and presents the correct IP address and parameters to the IP Stack.

Simple Network Management Protocol (SNMP)

SNMP traps are supported for drive and library events. SNMP management query functions are supported by using a standard Management Information Block (MIB).

Hyper Text Transfer Protocol (HTTP)

An embedded web server provides a management GUI for library management and query capabilities.

Secure Socket Layer (SSL)

The tape library supports SSL, a protocol for transmitting private documents through the internet.

Key Management Interoperability Protocol (KMIP)

Used for communicating with the IBM® Security Lifecycle Key Manager and other security key management software.

Simple Mail Transfer Protocol (SMTP)

The tape library supports SMTP for sending email alerts.

Network Time Protocol (NTP)

The tape library supports NTP for external time-and-date synchronization.

Lightweight Directory Access Protocol (LDAP)

The tape library supports LDAP for centralized authentication.

Domain Name System (DNS)

The tape library supports DNS for flexible IP addressing.

Dynamic Host Configuration Protocol (DHCP)

The library supports DHCP for automatically providing an Internet Protocol (IP) host with its IP address and other related configuration information, such as the subnet mask and default gateway.

Host requirements

The drive is supported by a wide variety of servers, operating systems, and adapters. Many ways to determine the servers and software that supports this drive are available.

- [SAS interface](#)
LTO drives can have a dual port 12 Gbps SAS (Serial Attached SCSI) host interface. The SAS interface speed varies with the generation of the LTO drive. For more information on SAS speed of different generation drives, see [Supported tape drives](#).
- [Fibre Channel interface](#)
The fibre channel interface varies with the generation of the drive.
- [Compatible Configurations](#)
- [HBA requirements](#)
The library requires attachment to supported SAS or FC HBAs.
- [con_3572_supported_device_drivers.html](#)
- [Application software](#)

SAS interface

LTO drives can have a dual port 12 Gbps SAS (Serial Attached SCSI) host interface. The SAS interface speed varies with the generation of the LTO drive. For more information on SAS speed of different generation drives, see [Supported tape drives](#).

A drive with a SAS interface can be linked directly to controllers. SAS is a performance improvement over traditional SCSI because SAS enables multiple devices (up to 128) of different sizes and types to be connected simultaneously with thinner and longer cables. Its full-duplex signal transmission supports up to 12.0 Gbps. In addition, SAS drives can be hot-plugged.

SAS drives auto-negotiate speed. There are no configurable topologies thus no feature switches associated with SAS.

Fibre Channel interface

The fibre channel interface varies with the generation of the drive.

Fibre Channel technology combines the best features of traditional input/output interfaces with the best features of networking interfaces. The technology offers a transport mechanism for delivering commands, and provides high performance by allowing processing to be done in the hardware.

The Fibre Channel connector is a Small Form-Factor Pluggable (SFP), dual-port, multimode optical transceiver that consists of an industry standard duplex LC-type connector.

You can establish Fibre Channel connections between Fibre Channel ports that are in the tape library, one or more servers, and the connecting network. The network can consist of such elements as switches, hubs, bridges, and repeaters.

Compatible configurations

For a comprehensive list of compatible configurations, see the IBM® Interoperation Center (SSIC) at <http://www-03.ibm.com/systems/support/storage/ssic/interoperability.wss>.

Compatible servers and software

Note: These attachments can change throughout the lifecycle of the product.

For complete IBM storage interoperability information for the tape drive in a storage area network (SAN) configuration, see the IBM Interoperation Center (SSIC) at <http://www-03.ibm.com/systems/support/storage/ssic/interoperability.wss>. The SSIC has details on supported operating systems, servers, switches, and adapters.

Note:

1. IBM does not provide application software with the drive. To order software, contact your IBM sales representative, IBM Business Partner, or an independent software provider.
2. If you attach your drive to a server with non-IBM software, contact your software vendor for a matrix of compatible hardware, software, firmware revisions, and adapters.

Note: Dependent on the use of Random or Sequential Mode, the drive might require an HBA with multiple LUN supports. Also, multiple LUN supports must be enabled on the host computer. When multiple LUN supports are not enabled, the host computer can see the tape drive, but not the library.

HBA requirements

The library requires attachment to supported SAS or FC HBAs.

The library requires attachment to supported SAS or FC HBAs. See [Host connectivity](#).

	Static Sensitive Risk of damage to devices • A discharge of static electricity damages static-sensitive devices or micro circuitry. • Proper packaging and grounding techniques are necessary precautions to prevent damage.

Follow these general guidelines.

- Check with a system administrator before the host computer is powered off.
- For a SAS library, confirm availability or install a SAS HBA that supports multiple LUNs.
- For a direct-attach Fibre Channel library, confirm availability of installation of an FC HBA.
- For connection of a Fibre Channel library through a compatible switch, verify that sufficient ports are available.

Persistent binding to ensure SCSI ID assignment

When a server is booted, devices are discovered and assigned SCSI target and LUN IDs. It is possible for these SCSI assignments to change between boots. Some operating systems do not guarantee that devices are always allocated the same SCSI target ID after rebooting. Also, some software depends on this association, so you do not want it to change. The issue of SCSI ID assignment is addressed by persistent binding.

Application software

Industry-leading compatible software offerings provide storage and tape management software for the LTO tape drives. Supporting software and applications must be obtained separately from IBM®, IBM Business Partners, or independent software vendors (ISV). For a list of compatible software and additional information, refer to the ISV Matrix that is available at [Independent Software Vendor \(ISV\) matrix for 3592 and LTO](#).

To download a free version of Adobe Reader, go to the [Adobe Acrobat website](#).

Optional features

More features that are available for your library.

To order extra features, contact your IBM® sales representative or Business Partner.

Table 1. Optional features

Feature Code	Description
1411	Fibre wrap tool
1412	SAS wrap tool
1682	Path Failover
1899	First Power Supply
1900	Extra Power Supply
7002	Rack Mount Kit
8106	Right Side Magazine
8109	Left Side Magazine
5500	Mini-SAS/Mini-SAS 4x Interposer Enables attachment of a single tape drive to a host with HD applications (from HD HBA with SFF-8644 to Drive with SFF-8088 connector).
5502	2 m Mini-SAS/Mini-SAS 1x Cable Enables attachment of a single tape drive to a host with HD applications (from HD HBA with SFF-8644 to Drive with SFF-8088 connector).
5507	4 m Mini-SAS HD/Mini-SAS 1x Cable Enables attachment of a single tape drive to a host with HD applications (from HD HBA with SFF-8644 to Drive with SFF-8088 connector).
5509	3 m Mini-SAS HD/Mini-SAS 2x Cable Enables attachment of one or two tape drives to a host with HD applications (from HD HBA with SFF-8644 to Drives with SFF-8088 connectors). This cable fans out to enable connection of two devices to one HBA port.
5900	Encryption Configuration
AGK1	10 m OM3 fiber Cable (LC)

Feature Code	Description
AGK2	25 m OM3 fiber Cable (LC)
AGKB	3 m Mini-SAS HD/Mini-SAS HD 1X Cable
AGKC	3 m Mini-SAS HD/Mini-SAS HD 2X Cable
AGKD	1.5 m Mini-SAS HD/Mini-SAS HD 1X Cable
AGKF	LTO 6 HH Fibre Channel Drive (Type: LTO Ultrium 6-H)
AGKG	LTO 6 HH SAS Drive (Type: LTO Ultrium 6-H)
AGKH	LTO 6 FH Fibre Channel Drive (Type: LTO Ultrium 6)
AGKJ	LTO 7 HH Fibre Channel Drive (Type: LTO Ultrium 7-H Fibre Channel)
AGKK	LTO 7 HH SAS Drive (Type: LTO Ultrium 7-H SAS)
AGKL	LTO 7 FH Fibre Channel Drive (Type: LTO Ultrium 7)
AGKM	LTO 8 HH Fibre Channel Drive (Type: LTO Ultrium 8-H Fibre Channel)
AGKN	LTO 8 HH SAS Drive (Type: LTO Ultrium 8-H SAS)
AGKP	LTO 8 FH Fibre Channel Drive (Type: LTO Ultrium 8)
AGLA	LTO 9 HH Fibre Channel Drive (Type: LTO Ultrium 9-H Fibre Channel)
AGLB	LTO 9 HH SAS Drive (Type: LTO Ultrium 9-H SAS)
AGLC	LTO 9 FH Fibre Channel Drive (Type: LTO Ultrium 9 Fibre Channel)
AGLD	LTO 9 FH SAS Drive (Type: LTO Ultrium 9 SAS)
8002	Cleaning Cartridge L1 UCC
8905	Ultrium 9 Data Cartridges (5-pack)
8806	LTO 8 Data Cartridges (5-pack)
5521	LTO M8 Data Cartridges (20-pack)
8706	LTO 7 Data Cartridges (5-pack)
8605	LTO 6 Data Cartridges (5-pack)
9800 - 9847	Power cords
9848	Rack Power Distribution Unit (PDU) power cord

Installing

Use this section to follow the procedures to install and configure your library.

Table 1. Installation Precautions

     	Product Weight Caution: The weight of this part or unit is between 18.1 and 33.6 kg (40 and 74 lb). It takes two persons to safely lift this part or unit. (C009) Caution: The weight of this part or unit is between 33.6 and 46.3 kg (74 and 102 lb). It takes three persons to safely lift this part or unit. (C010) Risk of personal injury Before a module is lifted or moved - Observe local health and safety requirements and guidelines for manual material handling. - Remove all tapes to reduce the weight and to prevent cartridges from falling into the robotics path and damaging the library. - Remove all tape drives to reduce the weight. - Obtain adequate assistance to lift and stabilize the module during installation or removal. Risk of damage to devices When a module is placed into or the module is removed from a rack - Extend the rack's leveling jacks to the floor. - Ensure that the full weight of the rack rests on the leveling jacks. - Install stabilizing feet on the rack. - Extend only one rack component at a time. - Do not expose the library to moisture. - Do not place a module on either the ends or sides as this action might cause damage.

- [Unpacking the Base Module and Expansion Modules](#)

Procedure for safely unpacking the Base and Expansion Modules.

- [Identifying Library Module components](#)

Use the packing slip that is included with your module to identify the module components.

- [Installing a tabletop module](#)

Installation of a one module library (Base Module only) can be completed without special hardware.

- [Removing inner foam from base module](#)

There is a foam packing placed inside each base module to protect the accessor from damage while shipping. When installing the modules, this foam packing must

be removed.

- [Preparing top and bottom modules](#)

Use these steps to prepare the top and bottom modules for installation.

- [Installing modules in a rack](#)

Rackmount installation procedure.

- [Aligning and connecting modules](#)

Aligning the modules ensures that the accessor can move freely between the modules.

- [Installing a tape drive](#)

Tape drives come already installed in the library modules.

- [Connecting cables](#)

Procedures to connect Fibre Channel, SAS, USB, and Ethernet cables.

- [Powering on the library](#)

Steps to power on the library.

- [The Initial Setup process](#)

When you turn on the library for the first time, the Initial Setup process starts automatically. Click Next to start the process.

- [Initial configuration and customization](#)

After the physical installation and initial setup by using the Operator Panel is completed, an administrator can log on to the Management GUI to complete the library configuration and configuration of any additional features.

- [Labeling and loading tape cartridges](#)

The library can power on without cartridges, but needs cartridges before it can complete data read and write operations, or any tests or operations that transfer cartridges.

- [Verifying the installation](#)

Verify that the library has the current firmware revision, and save the configuration settings. This action can be helpful if the library requires service.

- [Advanced library configuration](#)

To create and manage multiple logical libraries, utilize the advanced logical library function.

- [Verifying the host connection](#)

Procedure to verify the connection between the host computer and the library.

- [Registering for support notification](#)

Support notification registration provides email notification when new firmware levels are updated and are available for download and installation.

Unpacking the Base Module and Expansion Modules

Procedure for safely unpacking the Base and Expansion Modules.

Before any modules are unpacked, clear a work surface near the targeted rack or table for installation.

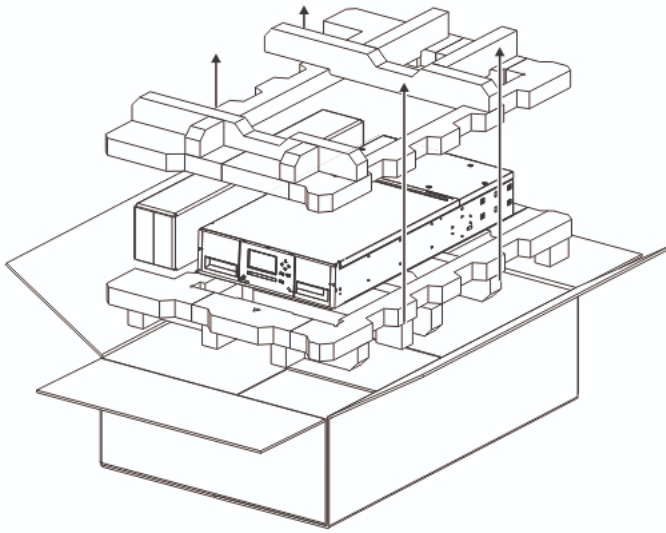
Attention: If the temperature in the room where the library operates varies by 15° C (30° F) from where the module was stored, allow it to acclimate for at least 12 hours before it is unpacked.

Unpacking a Base Module or Expansion Module

1. Before a module is opened or removed from the box, inspect the container for shipping damage.
2. If you notice any damage, report it to the shipping company immediately.
3. Remove the module from the box.

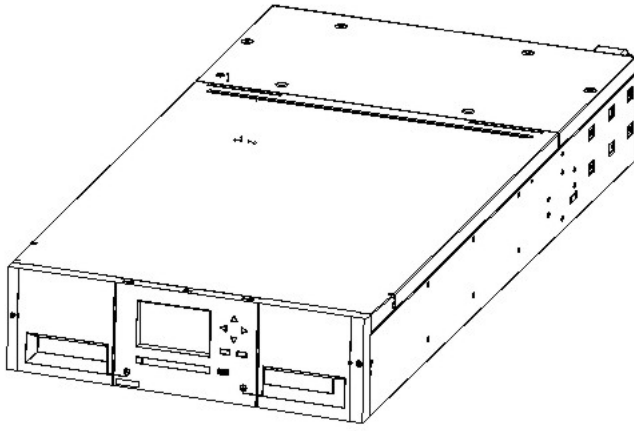
Important: Lift the module out of the box by the long sides, not by the display.

Figure 1. Removing the module from the box



4. Check that all components for assembling the module are in the box. See [Identifying Library Module components](#).

Figure 2. The module after removal from the box



Attention: Do not place a module on either the ends or sides as this action can damage the module.

Identifying Library Module components

Use the packing slip that is included with your module to identify the module components.

The TS4300 model 3555 is not shipped with a rack mount kit (IBM FC: 7002) and/or power cord (IBM FC: 9800-9848) unless it is ordered.

1. Locate one or more packing slips for your module.
2. Verify that you received each item that is listed on the packing slips.

Note: Order the power cord that matches the electrical requirements of the country or area. Power cord FC 9848 must be ordered with the rack mount kit if a rack-based power supply is used.

For SAS libraries, you must provide SAS cabling with the correct configuration for your HBA. For Fibre Channel libraries, you must provide one Fibre Channel cable for each tape drive. See [Optional features](#).

Installing a tabletop module

Installation of a one module library (Base Module only) can be completed without special hardware.

When unpacking a tabletop module, confirm that you received the following components:

1. Base Module
2. Power cord (IBM FC: 9800-9848 are not a part of the shipment unless ordered. For more information see, [Optional features](#).)
3. Feet kit

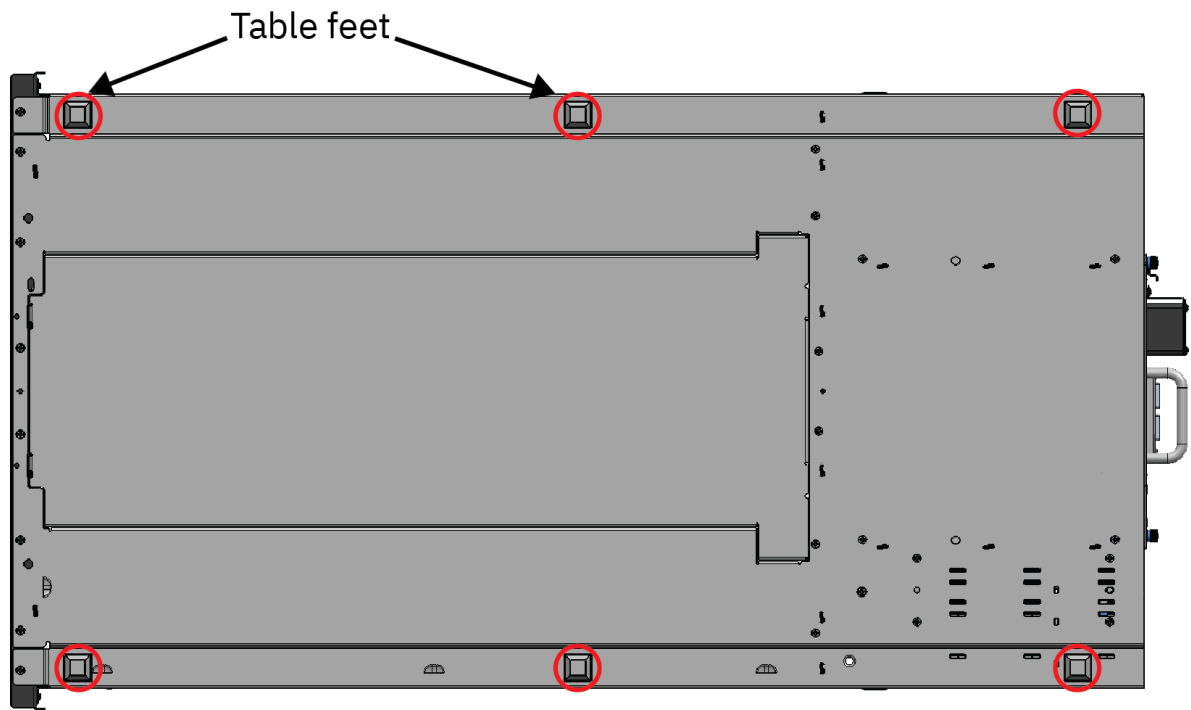
This procedure is meant to help the user in installing table feet to the module.

Important: Do not add feet to a rack-mounted module.

1. Remove the Base Module from the box.
2. Put the packing foam which came with the shipment on the table.
3. Place the module top side down over the packing foam on the table.
4. Add feet to the base of the module. See [Figure 1](#).

Attention: Ensure that the table feet does not cover any air holes at the base of the module.

Figure 1. Table feet



5. Place the module at the required location, with top side up. Ensure that it is level.
6. Open the cover and remove the foam packing from inside the enclosure. See [Removing inner foam from base module](#).
7. Replace the cover.
8. Plug in the power cord and the connecting cables.

Removing inner foam from base module

There is a foam packing placed inside each base module to protect the accessor from damage while shipping. When installing the modules, this foam packing must be removed.

Remember: The foam packing is only available inside a base module.
Follow these instructions to remove foam packing from a module.

1. Unlatch the top of the module by using your fingers or a small tool, one on each side of the lid, and press inward. When the lid opens, remove it by pulling it forward. See [Figure 1](#).

Figure 1. Unlatching the top of the module

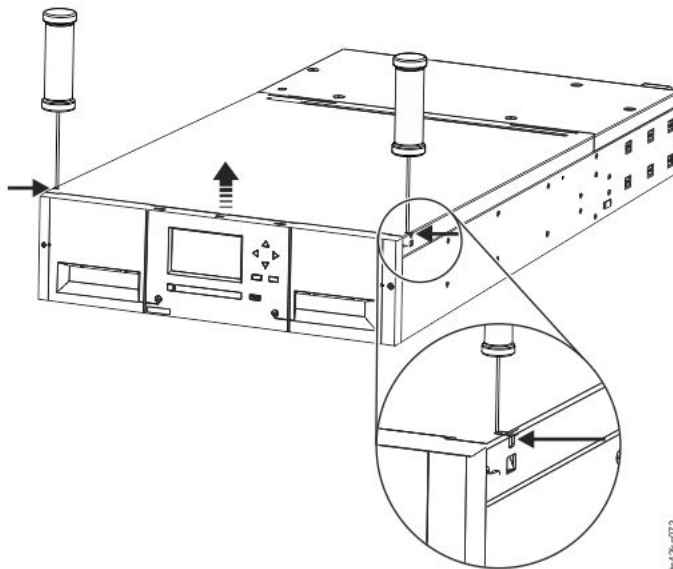
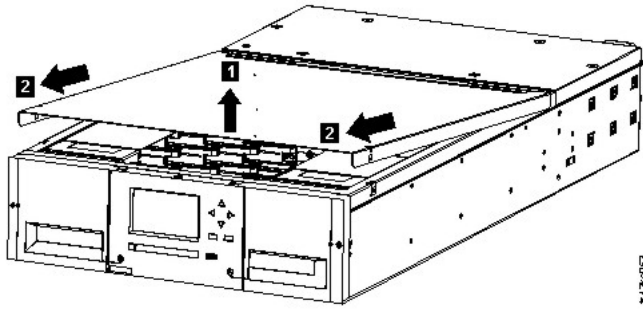
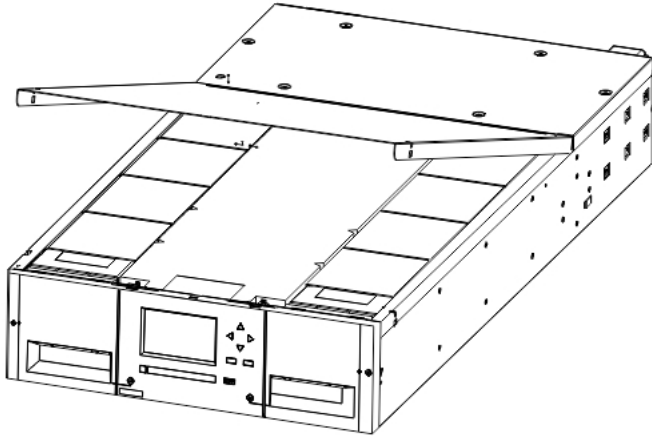


Figure 2. Removing the top of the module



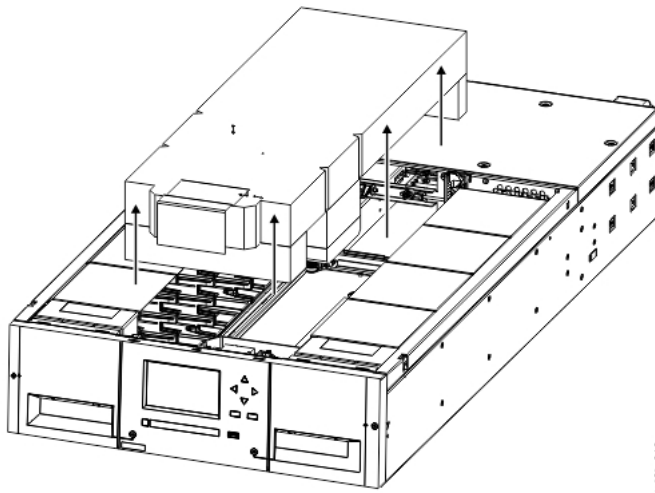
2. Remove the foam packing from the inside of the module.

Figure 3. The module is opened to show the foam packing.



3. After the packing is removed, the internal components are shown.

Figure 4. The foam packing is removed, and the internal components are shown - Base Module.



4. Install the top cover if you do not plan to add modules above this module.
5. Save the packaging materials for future use.
6. If you are adding extra modules, go to [Preparing top and bottom modules](#).

Preparing top and bottom modules

Use these steps to prepare the top and bottom modules for installation.

Skip this step if you are installing a Base Module only without an Expansion Module.

The Base Module has a removable top and bottom covers.

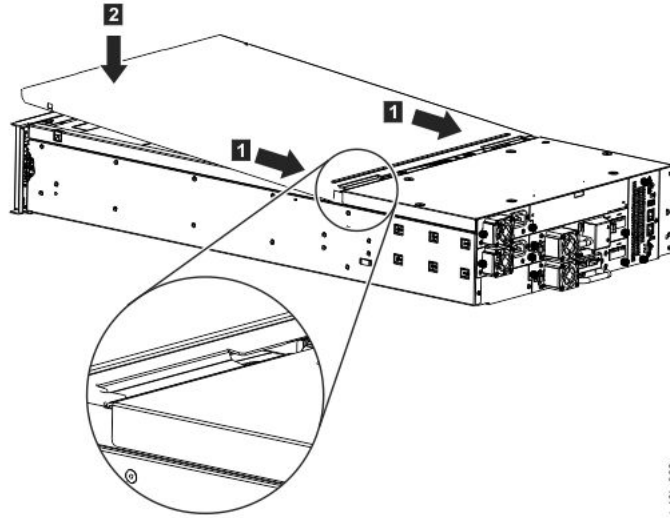
Installing Expansion Modules above the Base Module

If you are installing one or more Expansion Modules above the Base Module, move the top cover from the Base Module to the Expansion Module that is installed at the top of the library.

To move the library top cover plate from the Base Module to an Expansion Module

1. Remove the library top cover plate from the Base Module. See Step 5 in [Unpacking the Base Module and Expansion Modules](#).
2. Install the top cover on the Expansion Module that is installed on the top of the library.
 - a. Place the Expansion Module on a work table.
 - b. With the front of the top cover raised approximately 12 cm, engage the rear of the cover at the Expansion Module pivot point at the back of the opening.
 - c. Lower the front of the top cover until the latches engage on both sides.

Figure 1. Lowering the front of the top cover



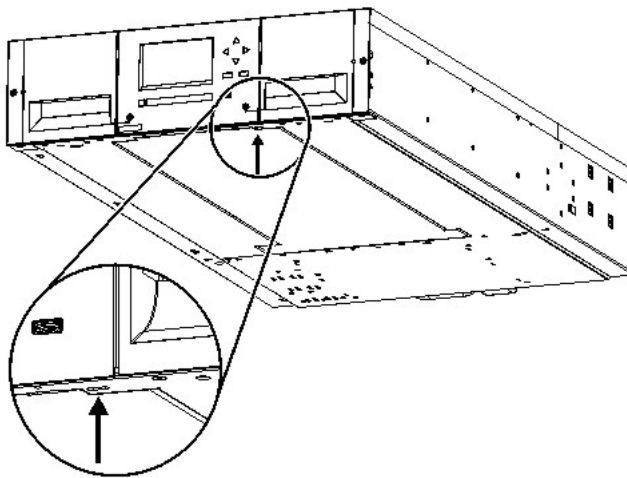
Installing Expansion Modules below the Base Module

If you are installing one or more Expansion Modules below the Base Module, move the bottom cover from the Base Module to the Expansion Module that is installed at the bottom of the library

To move the library bottom cover plate from the Base Module to an Expansion Module

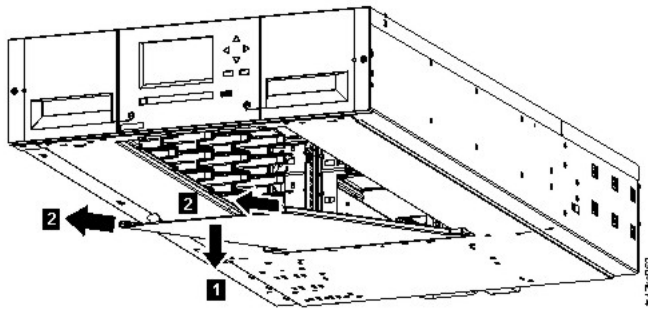
1. Remove the library bottom cover plate from the Base Module.
 - a. Place the Base Module on a work table.
 - b. Lift the unit front end by about 16 cm (use unit rear as a pivot edge).
 - c. Support the bottom cover with one hand. Insert a small flathead screwdriver or Torx screwdriver into the hole and slide about 4 mm sideways to the left to unlock the spring loaded lock. See [Figure 2](#).
- Important: Do NOT turn the module upside-down to complete this step.

Figure 2. Unlocking the spring loaded lock



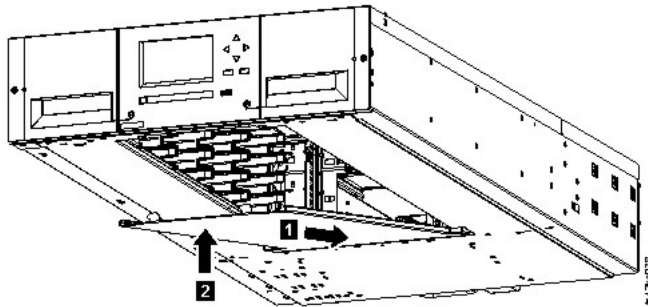
- d. Lower the cover front end by about 10 cm (1) and pull gently forward (2) to disengage from the pivot point at the unit center.

Figure 3. Removing the cover



2. Install the library bottom cover plate to an Expansion Module.
 - a. Place the Expansion Module on a work table.
 - b. Lift the unit front end by about 16 cm (use unit rear as a pivot edge).
 - c. Insert the bottom cover at the center
 - d. Lift the cover front edge until hard stop and it locks in at the unit front. The bottom cover fits only one way.

Figure 4. Lifting the cover and locking it



Installing modules in a rack

Rackmount installation procedure.

Modules are easy to install in racks that are compliant to the EIA 310A Standard, when at least 1 meter deep. You need a #2 Phillips screwdriver for this process.

Note: Install modules from the bottom to the top. Refer to [Structure and supported library configurations](#) for the correct configuration of Base and Expansion Modules. To locate the rail locations when multiple modules are installed.

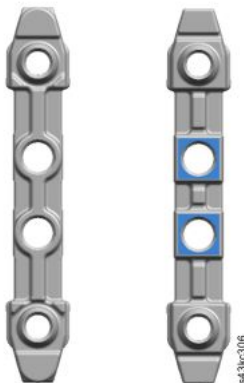
1. Locate the bottom of the lowest full U where the lowest module is installed.
2. Continue identifying the locations for any additional module 3U higher.

To install the rails into the rack, starting from the lowest rack location.

- a. Locate the four universal rack connectors, four Philips screws, and two rackmount rails (LH and RH).

Note: The universal rack connectors have two sides, for round hole and square hole racks. The square-hole side might be painted.

Figure 1. Universal rack connector



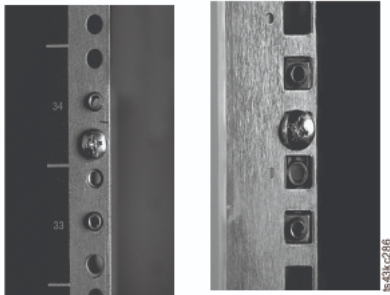
- b. On the inside of the racks, facing out, mount the connectors at the appropriate height to the right and left rack posts. Mount them in the middle hole of the height unit (the middle of a height unit is the hole between two wide and neighboring division bars) in both front and back. The four screw holes must align with the holes on your rack. If they do not, the blocks are not in the correct location. See [Figure 2](#) and [Figure 3](#).

Note: If the connectors are installed incorrectly, the screws on the connectors do not match the holes on the frames. The circles in the graphics highlight the mismatch.

Figure 2. Incorrect connector locations



Figure 3. Correct connector locations



- c. Repeat step b on the right and left rack posts in the rear of the rack.
- d. Mount the LH Rackmount rail to the connectors. See [Figure 4](#).
- e. Repeat step d with the RH Rackmount Rail.

Figure 4. Mounting the rails to the connectors

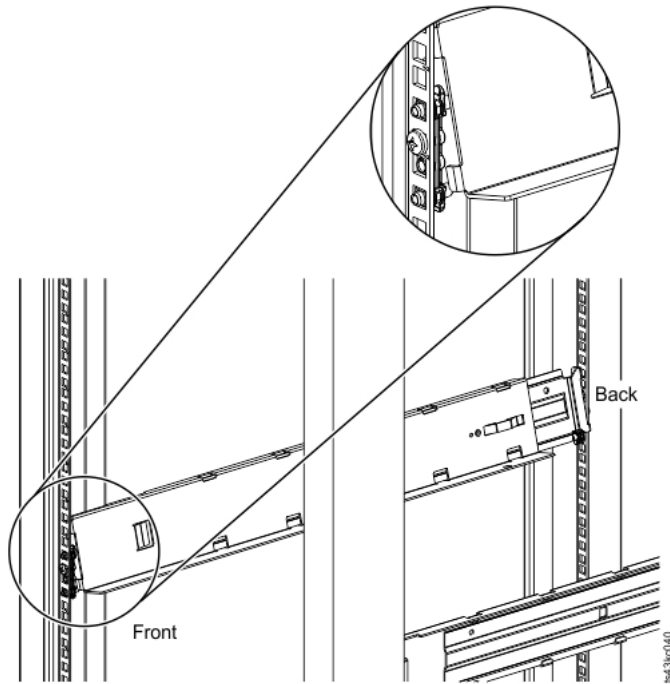
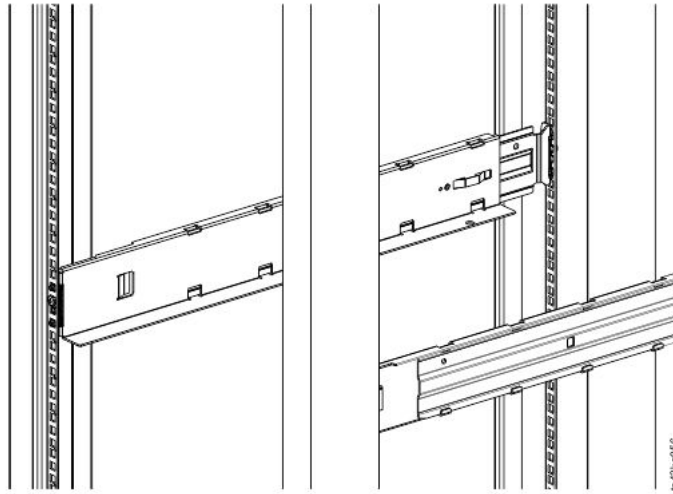
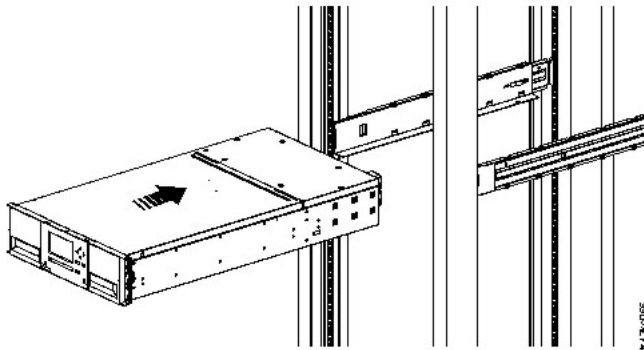


Figure 5. Side rails installed



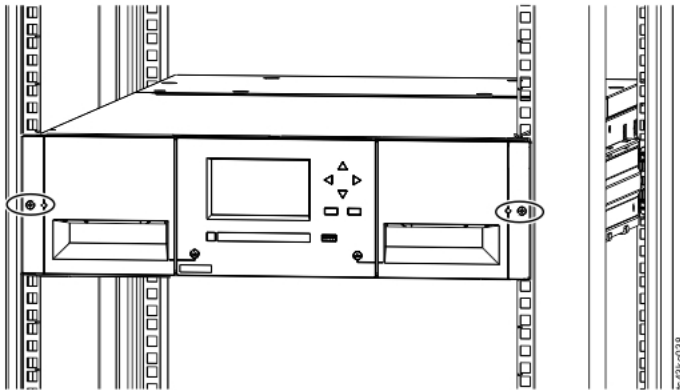
3. Place the library at the front of the rack on the support angles of the rails and push it into the rack to the back stop.

Figure 6. Sliding the library into the rack



4. If you are installing multiple modules, verify that this module is installed directly above or below its adjacent module and is contained within the correct 3U volume. Remove the tape that is covering the alignment pin lock/unlock lever on the rear of each module. The gap between modules must be less than 4 mm.

Figure 7. Library in the rack



Important: Each module must be on its own rails.

5. With a Phillips screwdriver, loosely screw the module to the front of the rack, one screw on each side. See the circled areas in [Figure 7](#).
6. Align the module as needed. Then, tighten the screws on each side of the module. See [Aligning and connecting modules](#).
7. Repeat steps 2 - 6 to install the rest of the modules into the rack.

Aligning and connecting modules

Aligning the modules ensures that the accessor can move freely between the modules.

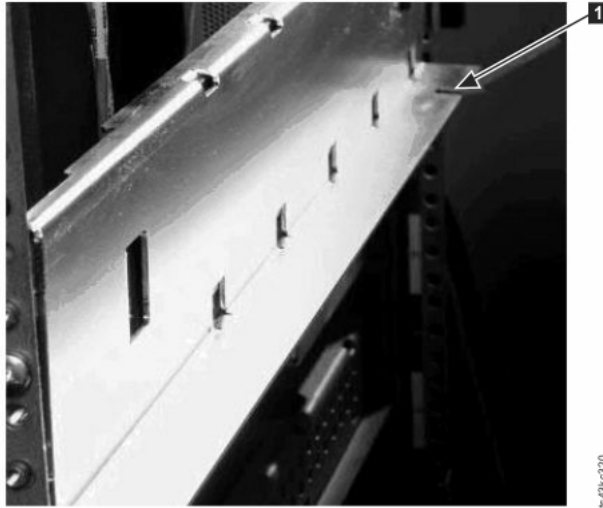
Skip this step if the library does not have Expansion Modules.

Aligning the modules ensures that the accessor can move freely between the modules. The library cannot operate unless the alignment mechanisms of the upper modules are in the locked position, and the alignment mechanism of the lowest module is unlocked.

1. From the front of the library, loosen the screws on each of the modules where they are attached to the rails two full turns.

2. From the back of the library, starting with the bottom pair of modules, align each module with the module below it. Repeat for each pair of modules. Refer to [Figure 3](#).
 - a. Move the alignment lever of the upper of the pair of modules to the locked or engaged position. If you encounter resistance, adjust the position of the upper module so the pin in the alignment mechanism moves into the mating hole in the lower module. If you still encounter resistance, check to see if the rack rails are installed correctly. Check that the hole for the alignment pin is on the left rail (looking from the front) toward the back of the rack. See 1 in [Figure 1](#).

Figure 1. Hole for alignment pin



Note: If a blue alignment lever lock is attached to the rear of the module, slide it to the left, then move the alignment lever. The lever lock has an internal spring, so hold it while the alignment lever is moved, and it automatically springs back into place after the lever is moved. See [Figure 2](#).

Figure 2. Alignment lever lock

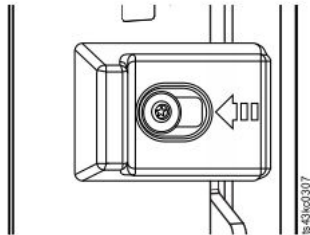


Figure 3. Alignment lever locked or engaged to lower module

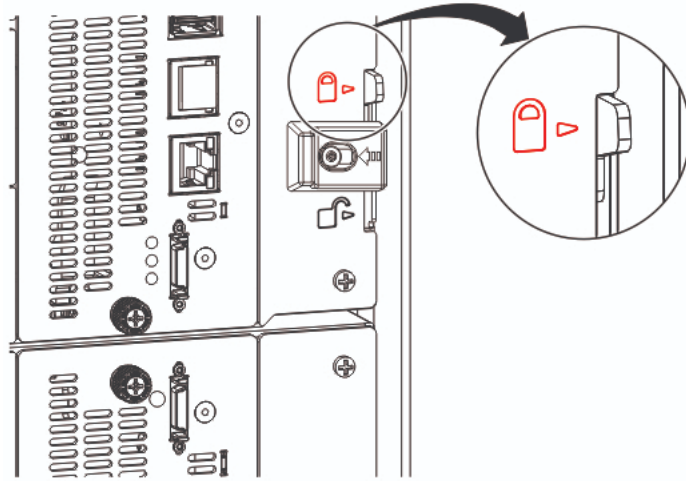
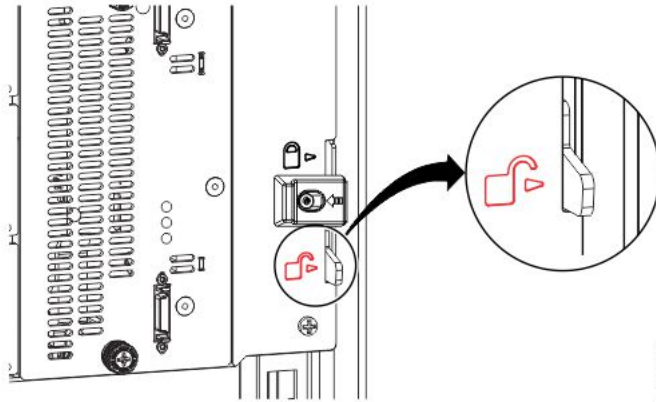
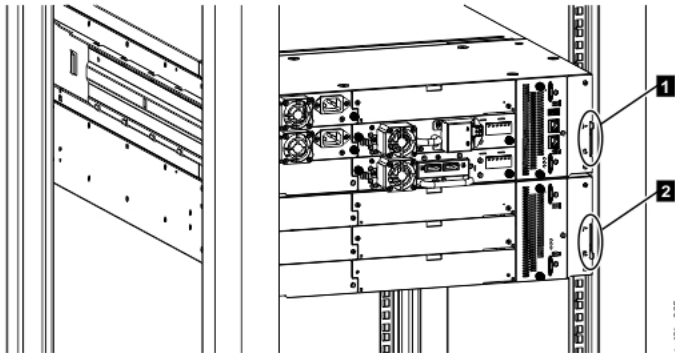


Figure 4. Alignment lever unlocked or disengaged



3. Verify that the lowest module in the library has its alignment lever is in the unlocked or disengaged position.

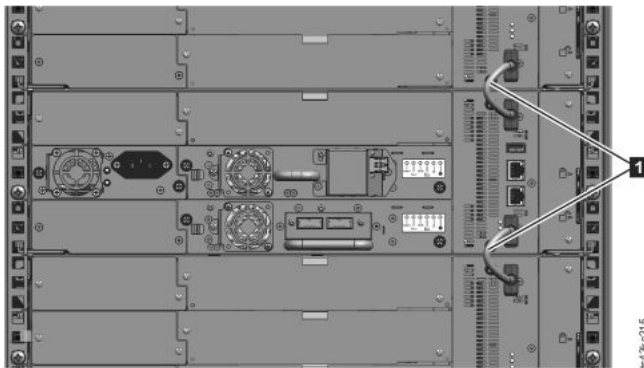
Figure 5. Two modules in rack, seen from the rear



1	Locked
2	Unlocked

4. From the front of the library, tighten the Philips screws on each of the modules to secure the modules to the rack.
5. From the back of the library, connect the modules of each pair to its adjacent module by using the expansion interconnect cables (1) as shown in [Figure 6](#).
Note: The top module's top connector and the bottom module's bottom connector has nothing plugged into them.

Figure 6. Connected modules



Installing a tape drive

Tape drives come already installed in the library modules.

Remember:

- Half-height tape drives can be installed in any drive bay in a module.
- Full-height tape drives must be installed in the lowest two bays of a module. Installing a full-height drive in the top two bays of a module is not supported.

For detailed instructions, see [Adding, removing, or replacing a tape drive](#).

Connecting cables

Procedures to connect Fibre Channel, SAS, USB, and Ethernet cables.

Connecting Fibre Channel cables

1. Remove the FC port caps if necessary. Attach one end of the FC cable to port 0 on the tape drive.

Figure 1. Full-height FC dual port

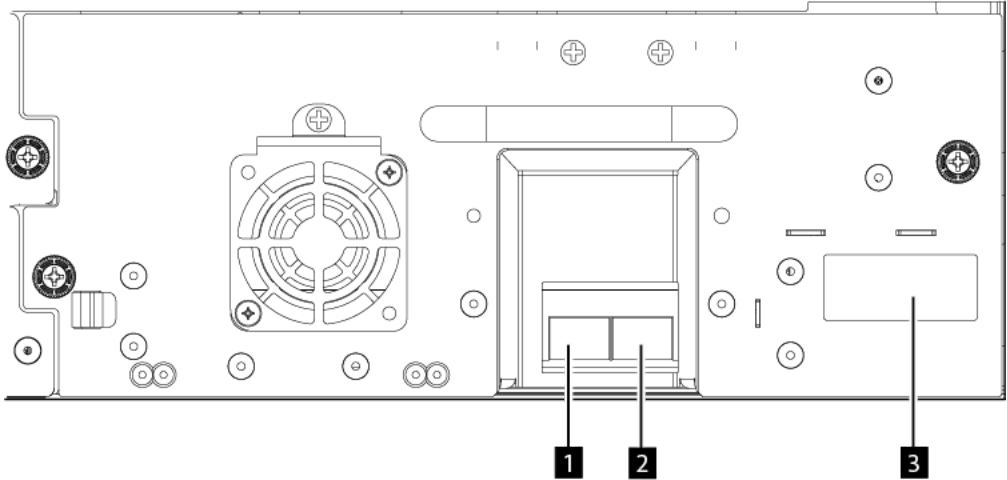


Table 1. Full-height FC dual port

Number	Description
1	FC port 0
2	FC port 1
3	Drive sled indicators (see Figure 1)

Figure 2. Half-height FC single port

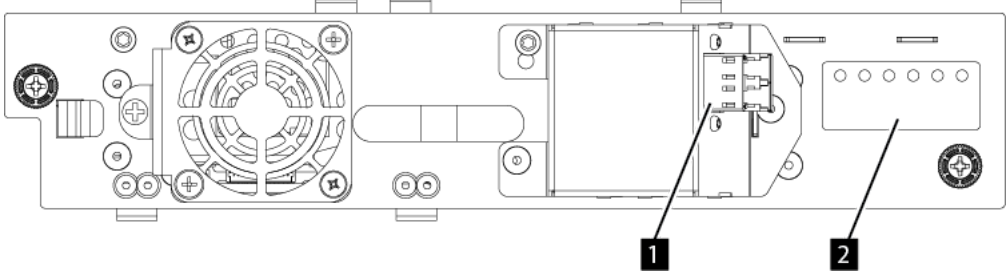


Table 2. Half-height FC single port

Number	Description
1	FC port 0
2	Drive sled indicators (see Figure 1)

2. Attach the other end of the FC cable to a switch or HBA.
3. Repeat the same process with port 1 if you have a dual port drive.

Connecting SAS cables

1. Attach the end of the SAS cable into the connector on the HBA. If you are using a SAS fanout/Interposer cable, the end of the cable with only one connector must be plugged into the HBA.
2. Connect the drive end of the cable.
- If you are using a cable with a single connector on each end, attach the other end into the connector on the tape drive.
 - If you are using a SAS fanout/Interposer cable, attach one mini-SAS connector into the connector on each tape drive. The unused ends of the SAS fanout/Interposer cable are single channel and not suitable for use with disk arrays. Use the other ends to connect tape drives, or coil and secure them to the rack to minimize stress on the connectors.

Figure 3. Half-height SAS dual port

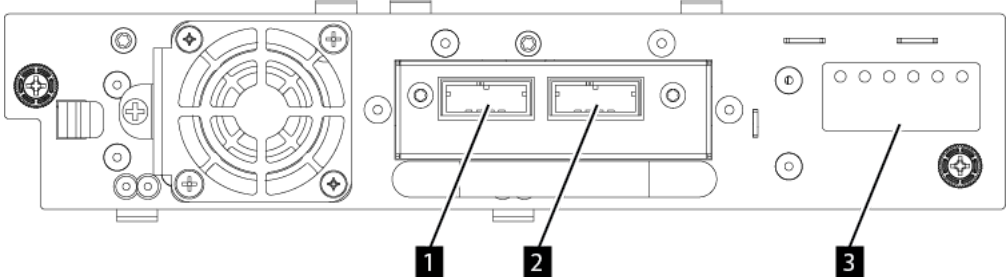


Table 3. Half-height SAS dual port

Number	Description
1	SAS port 0

Number	Description
2	SAS port 1
3	Drive sled indicators (see Figure 1)

Important: SAS signal rates require clean connections between the HBA and tape drive. Do not use adapters or converters between the HBA and the tape drive. For reliable operation, use a SAS cable length of maximum 6 meters for 6 Gbps or slower SAS speed. For 12 Gbps SAS speed, use a SAS cable length of maximum 3 meters.

Connecting USB cables

Two USB ports are on the library, one in the front and one in the rear. USB connections are used by service personnel for diagnostic and service procedures. Attach one end of the USB cable to your notebook or other device and the other end to the front or rear USB port of the library.

Important: USB cable lengths of more than 3 meters are NOT supported for the front or rear USB ports.

Connecting Ethernet cables

To use the Management GUI, connect an Ethernet cable from the bottom Ethernet port on the Base Module controller to your network. See [Rear panel](#) for the location of the Ethernet ports.

Remember: Ethernet port A (bottom Ethernet port) is the primary port. The second Ethernet port, Port B, is for redundancy.

Powering on the library

Steps to power on the library.

1. Plug the power cables into the power connectors on each module and into power outlets.

Notes:

- The library has dual redundant power supplies. To increase redundancy, plug each power cord into a different AC power circuit.
- A power supply is required in expansion modules if drives are installed.

2. Power on the library by pressing Power on the Base Module just below the Operator Panel and hold for 5 seconds. See [Front panel](#) for the location of the Power button. When the library is powered on, it
 - a. Inventories the tape cartridges in the magazines,
 - b. Checks the firmware version on all modules,
 - c. Configures the tape drives.
 - d. Confirms the presence of the existing modules,
 - e. Searches for any new modules.
 - f. When the library is powered on for the first time, the Initial Setup starts. See [The Initial Setup process](#).

The Initial Setup process

When you turn on the library for the first time, the Initial Setup process starts automatically. Click Next to start the process.

The wizard guides you through setting the Internal IP range, setting library network configuration, configuring date and time, and setting the administrator PIN. You can skip items and stop the wizard at any time. After you configure the network settings, you can start the wizard from the Management GUI to complete more configuration items.

Notes on navigation and entering data into the Operator Panel

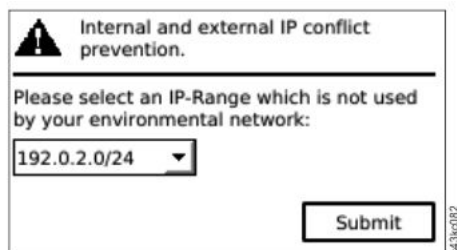
- The arrow keys on the front panel are used to select numeric and alphanumeric characters and symbols. Capital and lowercase letters, numbers, and punctuation are available to use.
- The right button under the arrows is Enter, which is pressed before you enter text.
- The left button under the arrows is Back/Return, which is used to delete entries.

See [Figure 1](#).

When the library starts up for the first time, the initial setup automatically begins.

1. Enter the IP address for your library. See [IP range selection](#).

Figure 1. IP address selection



2. Press Enter to unlock the Operator Panel.
Note: If you wait too long to make your selection, the unit auto calibrates. The auto calibration finishes, then returns you to the login screen.
3. When you are logged in, the initial configuration process begins with Network Settings.
4. Follow the prompts to set date and time, and administrator PIN.
5. When the initial setup is complete, the display returns to the Operator Panel main screen.

To check your configuration at any time, go to Configuration > Initial System Setup on the Operator Panel. On the Management GUI, go to Library.

Initial configuration and customization

After the physical installation and initial setup by using the Operator Panel is completed, an administrator can log on to the Management GUI to complete the library configuration and configuration of any additional features.

Upon the first login with the user role `administrator` and password: `adm001`, you must change your password. Your new password must have these characteristics:

- 8 characters long
- At least one lowercase alphabet character
- At least one uppercase alphabet character
- At least one numeric character
- No more than two consecutive characters

The Initial Configuration Wizard guides you through basic configuration settings.

The library has many features to customize it for your organization. Go to [Locating Management functions](#) to customize your library with these features.

- Enabling or disabling the I/O station.
- Naming the library with the Manage Logical Library function.
- Creating or managing Logical Libraries. See [Library sharing](#) for information.
- Selecting Random or Sequential Mode. See [Random and Sequential Logical Library modes](#) for information.
- Enabling and configuring SNMP network management.
- Setting up email event notification.
- Setting up encryption.
- Configuring date and time.
- Enabling or disabling Library Auto Clean. See [Methods of cleaning drives](#).

Labeling and loading tape cartridges

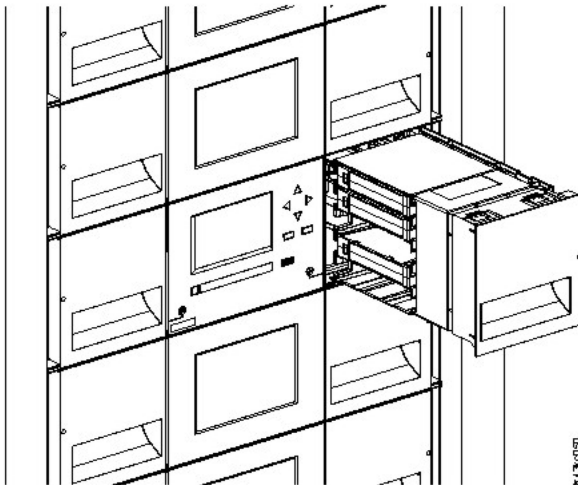
The library can power on without cartridges, but needs cartridges before it can complete data read and write operations, or any tests or operations that transfer cartridges.

Bar code labels are highly recommended in production environments to improve inventory time in the library and ease cartridge-handling processes outside the library. See [Bar code labels](#).

The I/O station

If the I/O station is enabled, you can use it to load cartridges into the library. Press the magazine button for less than 3 seconds. When the button LED starts flashing fast, pull out the I/O station. The right magazine will only pull out part way to reveal five slots.

Figure 1. Open I/O station seen from the left

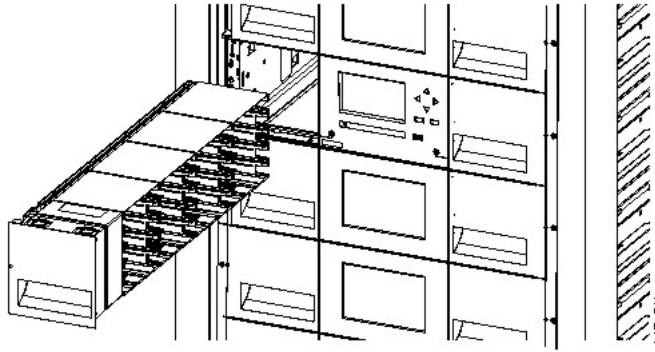


Bulk loading magazines

1. Unlock the magazine by pressing the magazine button for more than 3 seconds, wait for the button to flash fast and then pull out the magazine.
 - a. From the Operator Panel or Management GUI, select the module and then select Open Magazine. You can also press the release button on the front panel of the module to release the magazine.
 - b. Wait until the magazine is unlocked, and then pull out the magazine.

Note: Wait for the Operator Panel/Management GUI message to say the magazine is unlocked before the magazine is pulled out.

Figure 2. Magazine pulled out



2. Load the tape cartridges into the magazine.
Important: For libraries with serial numbers before 7800K0K, the slots of the lowest row of the bottom module are inaccessible and can contain a 4-slot I/O station only, so do not load cartridges into these slots.
3. Insert the magazine into the unit.
4. Push the magazine handle slowly until the magazine release latch snaps into place. The magazine locks into place.
Important: Push the magazine fully into place until the latch snaps into place.
5. Repeat steps 1 - 3 for each of the other magazines.

See [Accessing cartridges](#).

Verifying the installation

Verify that the library has the current firmware revision, and save the configuration settings. This action can be helpful if the library requires service.

Verify that the library has the current firmware revision. The library firmware revision is displayed at Library > Actions > Properties.

1. Verify library firmware and update if needed: Library > Actions > Update Library Firmware
2. Run Library Verify.
3. Save the configuration settings to a file on your computer from the Management GUI: Settings > Library > Advanced > Save Configuration File.
Having a backup of the library configuration is helpful when the library is recovering from a configuration error or needs service.

Advanced library configuration

To create and manage multiple logical libraries, utilize the advanced logical library function.

- [Overview](#)
Overview of advanced features such as multipath architecture, multiple logical libraries, and multiple control paths.
- [Library partitioning](#)
Libraries that contain at least two drives can configure two logical libraries. It is possible to configure up to 21 logical libraries in the library (up to the number of drives installed).

Overview

Overview of advanced features such as multipath architecture, multiple logical libraries, and multiple control paths.

Multipath architecture

The multipath architecture feature of this tape library allows Open Systems applications to share the robotics of the library. See [Library sharing](#).

The library features storage area network (SAN) ready multipath architecture. This architecture allows homogeneous or heterogeneous Open Systems applications to share the library's robotics without middleware or a dedicated server (host) acting as a library manager. The SAN-ready multipath architecture makes sharing possible by partitioning the library's storage slots and tape drives into logical libraries. Servers can then run separate applications for each logical library. This partitioning capability extends the potential centralization of storage that the SAN enables. Partitioning also provides investment protection if your application does not support the mixing of drive generations and media in the same logical library.

The multipath architecture of this library is designed to provide the capability to share the library robotics. The sharing is accomplished first by partitioning the library into multiple logical libraries (up to the number of drives installed). Then, each logical library is assigned its own separate and distinct drives, storage slots, and control paths. Input/output (I/O) slots are shared on a first-come-first-serve basis. This type of partitioning is designed to allow heterogeneous applications to share the library robotics independent of each other. Cartridges under library control are not shared between logical libraries, nor are they allowed to be moved between logical libraries. An example of heterogeneous sharing is a Microsoft Windows application that is using the drive and storage slots of one logical library, while a UNIX application uses the drive and slots of another logical library. See [Mixed drives](#).

Multiple logical libraries

A library can be partitioned into multiple logical libraries to enable simultaneous data backup and restore tasks from different applications. For example, you can create multiple logical libraries so that is processes

- Commands from Application 1 (about Department A) in Logical Library 1
- Commands from Application 2 (about Department B) in Logical Library 2
- Commands from Application 3 (about Department C) using Sequential Mode in Logical Library 3

In this configuration, the tape drives and cartridges of each logical library are dedicated to that library and are not shared among other libraries and applications. Commands that are issued by the applications travel to the library through unique control paths or sequential mode processing by the library. So, the data processing for Department A is confined to the tape drives and cartridges of Logical Library 1. Processing for Department B is confined to the tape drives and cartridges of Logical Library 2, and so forth.

For applications that do not support mixed drive types and media within the same logical library, partitioning the library into multiple logical libraries provides the capability to keep them separate. For example, you can partition the following tape drives and their media into multiple and separate logical libraries:

- LTO 8
- LTO 7
- LTO 6

Multiple control paths

With this tape library's multipath architecture, in addition to creating multiple logical libraries, you can configure any logical library to have more than one control path. A control path is a logical path into the library through which the library receives standard **SCSI Medium Changer** commands to control the library operations.

Note: No SCSI Medium Changer is configured when the logical library is enabled as Sequential Mode

Multiple control paths reduce the possibility of a failure in one control path to cause the entire library to become unavailable. Also, when you configure more control paths, more library-sharing configurations and options are possible. Access to the library is on a first-come, first-served basis. Each control path for a logical library can accept commands while the library is in use by another control path.

Multiple control paths for control path failover

This tape library offers an optional control path failover feature. See [Library sharing](#) and [Control path failover, Data path failover, and load balancing](#).

Use of the control path failover feature further reduces the possibility of a failure in one control path to cause the entire library to become unavailable.

The control path failover feature (feature code 1682) enables the host device driver to resend a command to a different control path for the same logical library.

Library partitioning

Libraries that contain at least two drives can configure two logical libraries. It is possible to configure up to 21 logical libraries in the library (up to the number of drives installed).

Partitioning of libraries

With full-height or half-height physical drives, physical numbering is bottom up for all drives. For example, if you replace a half-height drive in [Figure 1](#), the drives are still numbered 1-4. If you add a drive in any of the slots in between drives numbered 1 and 4, the physical numbering changes and is still numbered bottom up.

Important: A full-height drive can be installed in a module in the lower two slots only.

Configuration of a 1-logical library system

A one logical library system contains all drives present in any drive positions, and it contains all the slots.

Configuration of multiple logical libraries

A library with multiple logical libraries must have a drive for each logical library and at least five slots. Drives can be in any location in the library. It is best to have drives that are located near the slots that are assigned to the same logical library to minimize accessor movement and maximize performance.

SCSI element-addressing

A logical library assigns SCSI element addresses to drives, storage slots, I/O slots, and the accessor. For each element type (drive, storage, I/O), the SCSI element address can be viewed on the Management GUI.

While SCSI addressing follows the same method as physical location-numbering, this action depends on the Advanced Logical Library configuration.

Drive numbering is from bottom to top. Storage slot-numbering is from left magazine (front to back, bottom to top) to right magazine (back to front, bottom to top). IO slot-numbering is from bottom to top. Accessor is single number.

Note: When the number of drives in your library are reduced, update the logical library configuration. This action removes all event notifications that indicate a drive is missing.

Updating the logical library configuration might change the SCSI element addressing.

Verifying the host connection

Procedure to verify the connection between the host computer and the library.

To verify the connections between the host computer and the library

1. Install the application software and drivers that are compatible with the library. Backup software packages might require extra software or licensing to communicate with the robotics.
2. Verify the connection between the library and the host by using the host server's operating system utilities. Or, use the IBM® Tape Diagnostic Tool (ITDT) to verify the communication between library and host. See [IBM Tape Diagnostic tool \(ITDT\)](#).

See [Host connectivity](#) for compatible servers and software.

Registering for support notification

Support notification registration provides email notification when new firmware levels are updated and are available for download and installation.

To register for support notification, visit the web at: <http://www-01.ibm.com/software/support/einfo.html>.

Enter your user name and password on the [Library Configuration Form](#).

Note: Library firmware and tape drive firmware are verified and released together. When the latest firmware is updated, verify that all installed components such as the tape drive and library are at the latest levels noted on the Support website. Mixing different levels of library and tape drive firmware is not supported and might cause unpredictable results.

IBM® suggests that you update library and drive firmware when new levels become available. For instructions on updating library and drive firmware, see [Updating library and drive firmware](#).

Now you are ready to use your library.

Managing

Four user roles are described, and each user role has its specific functions.

- Administrator - This role provides access to the administrator functions on the Management GUI. There is a default administrator password **adm001** for the first login. The administrator password can be changed on the Local Users page.
- Monitor - This role allows access to library status information and does not allow access to configuration, maintenance, or operation features. Setting a monitor password restricts access to status information to only those users who know the monitor password. Passwords for the Monitor role can be set or changed by the administrator.
- Superuser - This role has same access rights as the Administrator role, except the ability to access the Local Users and Remote Authentication (LDAP Authentication and Kerberos Authentication) pages. In addition, it is possible to do cartridge moves and open magazines and I/O Stations. Passwords for the Superuser role can be set or changed by the administrator.
- Service - This role provides access to the service functions on the Management GUI. Passwords for the Service role can be set or changed by the administrator.

Notes:

- Monitor, Superuser, and Service user IDs must be enabled by the library administrator. These accounts are disabled by default.
- For a complete description of the menu items available to each user role, see [Management GUI functions and roles](#).
- [The Management GUI](#)
With the Management graphical user interface (GUI), you can monitor, configure, and operate most library functions from a web browser.
- [The Operator Panel](#)
With the Operator Panel, you can monitor, configure, and operate library functions from the library front panel.
- [Locating Management functions](#)
This table provides the menu navigation to assist with library setup and configuration.

The Management GUI

With the Management graphical user interface (GUI), you can monitor, configure, and operate most library functions from a web browser.

When possible, use the Management GUI as the primary library interface. The web interface provides access to more features, includes online help, and is intuitive to use.

Before the Management GUI can be used, you need to log in and configure the library network settings with the Operator Panel. This action can be done during Initial Setup. See [The Initial Setup process](#).

Logging in with the Management GUI

1. Open a supported web browser and enter the IP address of the library in the browser's address bar.
2. Type in the username (**administrator** or other administrator-created user) and the password. Click Login.
Note: For initial login, type **administrator** and the password **adm001**. You **must** change the password after the initial login. Your new password must have these characteristics:
 - 8 characters long
 - At least one lowercase alphabet character
 - At least one uppercase alphabet character
 - At least one numeric character
 - No more than two consecutive characters

Note: Only one person (on Operator Panel or Management GUI) can be logged in to the library at a time. If another person is already logged in when you try to log in, a dialog box appears asking if you want to log off the other user.

The Library main screen on the Management GUI

The library main screen is organized into the following regions:

Figure 1. Management GUI main screen

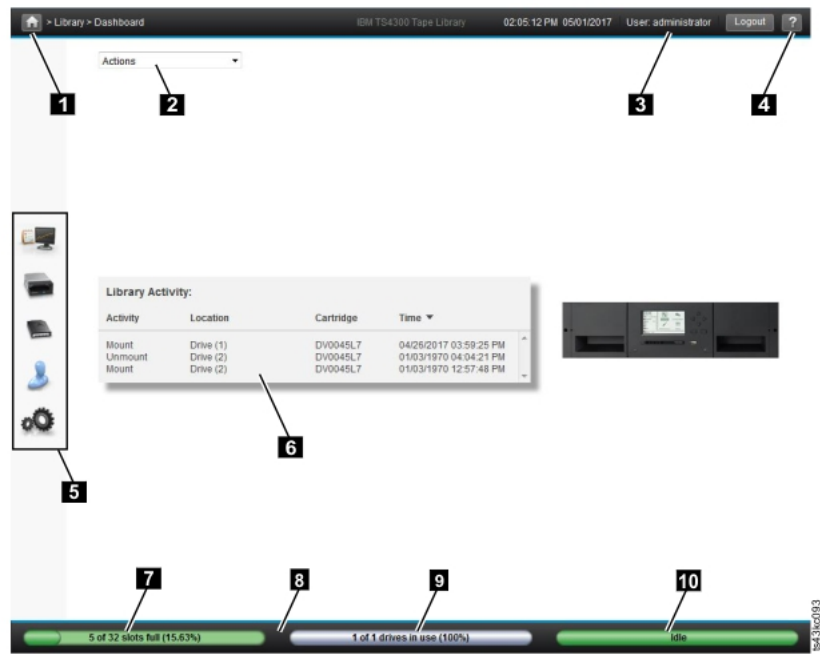


Table 1. Main screen elements

	Element
1	Home icon > Current Navigation
2	Actions > dependent on current navigation
3	User logged in
4	Help
5	Navigation Dock
6	Overview - dependent on current navigation
7	Physical Capacity
8	Status Bar
9	Drive Activity - navigate to Drive page for more details
10	Library Status

Tips:

1. For specific management function navigation, see [Locating Management functions](#).
2. For additional information, see online help pages in the Management GUI. The help pages are updated with firmware updates and often contain up-to-date technical details that might not be contained in this document. To access Management GUI help, click ? on the right side of the Management GUI top banner.
3. For information about user role permissions, see [Management GUI functions and roles](#).

Navigation Dock

Table 2. Navigation Dock

Navigation Dock Icons	Element	Extra menus
	Library	• Dashboard • Modules and Magazines • Logical Libraries • Events
	Drive	Drives and Ports
	Cartridges	Cartridges and Slots
	Access	• Local Users • Local Password Policies • LDAP Authentication • Kerberos Authentication
	Settings	• Library • Network • Notifications • Security

Status icons

Status icons indicate the following conditions.

Table 3. Status icons

Icon	Description
	The green OK icon indicates that the library is fully operational and that no user interaction is needed.
	The yellow exclamation point Warning icon indicates that user attention is necessary, but that the device can still finish most operations.
	The red X Error icon indicates that user intervention is needed and that the device can't finish some operations.

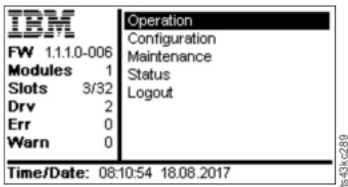
The Operator Panel

With the Operator Panel, you can monitor, configure, and operate library functions from the library front panel.

The Operator Panel has a Power button, an LCD display, six navigation buttons, and five LEDs. To use the Operator Panel, you must use the six navigation buttons (up/down, left/right, Enter, Back). The Operator Panel is **not** a touchscreen. See [Front panel](#) for the location of the navigation buttons.

Operator Panel screens

Figure 1. Operator Panel main screen



Operator Panel main screen layout

- Left Pane - Displays the library status (firmware revision, number of modules, number of slots, number of drives, number of errors, number of warnings).
- Center Pane - Provides access to operate, configure, and log out of the library and to view more status information (Operation, Configuration, Maintenance, Status).
- Bottom Pane - Displays more status information (library status, time/date, IPv4, or IPv6 address). The status pane displays one status information for 10 seconds and then switches to the next status item.

Table 1. Operator Panel menu tree

Operation	Configuration	Maintenance	Status	Logout
Move Cartridge from Drive to Home Slot	Initial System Setup	Library Tests	Network Settings	Logout
Move Cartridge	Date & Time	View Events	Library	
	Network Settings	Drive Service Logs	Drive	
	User Accounts	Download		
	Reset	Library Logs		
		Download		
		Drive Firmware		
		Upgrade		
		Library Firmware		
		Upgrade		
		LCD Adjustment		

The Operator Panel provides a subset of menu items that are compared to the full capability of the Management GUI. For the operations that are available on the Operator Panel, see [Locating Management functions](#).

Accessing the library with the Operator Panel

The Operator Panel can be accessed in two ways, with a PIN or without one.

1. If the Operator Panel screen saver is on, press Enter.
2. If no PIN is configured, press Enter.
3. If a PIN is configured, enter the PIN, then select Login and press Enter.

Status icons

Figure 2. Front panel LEDs

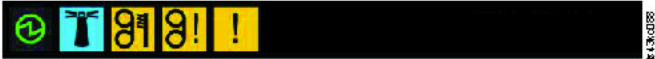


Table 2. Front panel LEDs

LEDs	Color	Descriptions
Ready	Green	Steady when power is on, flashing with tape Ready drive or library robotic activity.

LEDs	Color	Descriptions
Unit ID	Blue when activated	The unit identification (UID) LEDs are controlled by the user through the Maintenance > UID LED Control screen. The UIDs on the Operator Panel and base module back panel are activated and deactivated together. In addition, UIDs on drives and expansion module back panels can be activated separately. The UIDs are helpful for locating components of the library in a data center.
Clean	Amber	On, when a tape drive-cleaning operation is recommended.
Attention	Amber	Flashing if the library detected a condition for which user attention is necessary, but the library can still complete most operations.
Error	Amber	On, if an unrecoverable tape drive or library error occurs. A corresponding error message is displayed on the LCD screen. User intervention is required as the library is not capable of completing some operations.

Locating Management functions

This table provides the menu navigation to assist with library setup and configuration.

Table 1. Locating Management functions

Task	Menu Navigation	
	Operator Panel	Management GUI
Advanced settings	Not available with this interface	Settings > Library > Advanced
Auto Calibration	Not available with this interface	Settings > Library > Auto Calibration
Auto Clean	Not available with this interface	Library > Logical Libraries > Actions > Manage Logical Library (Expert Mode) See Methods of cleaning drives .
Cartridge, eject from a drive	Operation > Move Cartridge from Drive to Home Slot	Drives > Actions > Eject Cartridge from Drive
Cartridge Inventory, rescan	Not available with this interface	Cartridges > Actions > Inventory Library
Cartridges, list	Not available with this interface	Cartridges
Cartridges, move	Operation > Move Cartridge	Cartridges > Actions > Move Cartridges
Cartridges, graphical view	Not available with this interface	Cartridges > Actions > Graphical View
Certificates, create, backup, restore	Not available with this interface	Settings > Security > GUI
Cleaning, tape drive	Not available with this interface	Drives > Actions > Clean Drive See Methods of cleaning drives .
Configuration, save and restore	Not available with this interface	Settings > Library > Advanced
Configuration, reset	Not available with this interface	Settings > Library > Advanced
Configuration file, restore	Not available with this interface	Settings > Library > Advanced
Configuration file, save	Not available with this interface	Settings > Library > Advanced
Date and time, configure	Configuration > Date & Time	Settings > Library > Date and Time
Diagnostics, run Demo Mode	Maintenance > Library Tests	Library > Actions > Tests
Diagnostics, run Library Verify	Not available with this interface	Library > Actions > Tests
Diagnostics, run Drive test	Maintenance > Library Tests	Library > Actions > Tests
Diagnostics, run Slot to Slot exerciser	Not available with this interface	Library > Actions > Tests
Drive firmware, update	Maintenance > Drive Firmware Upgrade (requires FAT32 format USB drive)	Drive > Actions > Update Drive Firmware
Drives and Modules, reset the list	Not available with this interface	Settings > Library > Advanced
Drive status	Status > Drive	Drives
Drive service logs, download	Maintenance > Drive Service Logs Download (requires FAT32 format USB drive)	Drives > Actions
Drive, modify port settings	Not available with this interface	Drives > Actions
Email Notification	Not available with this interface	Settings > Notifications
Encryption, configure	Not available with this interface	Settings > Security > Encryption See Configuring Library Managed Encryption .
Encryption Connectivity Check	Not available with this interface	Settings > Security > Encryption
Encryption (LME) license key, add or delete	Not available with this interface	Settings > Library > Licensed Features
Encryption, reset	Not available with this interface	Settings > Security > Encryption
Factory/Manufacturing reset	Configuration > Reset > Full Factory Reset	Settings > Library > Advanced
Help	Not available with this interface	Click the ? at the upper right of the Management GUI screen. See 4 on Figure 1 .
Identifier light, turn On or Off	Not available with this interface	Library > Actions > Turn Identifier Light On or Off
Initial Setup	Configuration > Initial System Setup	Settings > Library > Initial Configuration Wizard
Inventory List	Not available with this interface	Cartridges
I/O Station, enable or disable	Not available with this interface	Library > Modules and Magazines > Actions > Enable or Disable I/O Station
I/O Station, open	See Accessing cartridges .	Library > Modules and Magazines > Actions > Unlock I/O Station
Kerberos Authentication, configure	Not available with this interface	Access > Kerberos Authentication
Key Path Diagnostic	Not available with this interface	Settings > Security > Encryption See Key Path Diagnostics .
LCD Adjustment	Maintenance > LCD Adjustment	Not available with this interface
LDAP Authentication, configure	Not available with this interface	Access > LDAP Authentication
Library firmware, update	Maintenance > Library Firmware Upgrade (requires FAT32 format USB drive)	Library > Actions > Update Library Firmware

Task	Menu Navigation	
	Operator Panel	Management GUI
Library logs, download	Maintenance > Library Logs Download (requires FAT32 format USB drive)	Library > Actions > Export Library Logs
Library logs, view or clear	Maintenance > View Event Ticket Logs	Library > Events > Actions
Library Information	Status	Library > Actions
Library Managed Encryption, configure	Not available with this interface	Settings > Security > Encryption See Configuring Library Managed Encryption .
Library Verify, run	Maintenance > Library Tests	Library > Actions > Tests > Library Verify
List of Known Drives and Modules, reset	Not available with this interface	Settings > Library > Advanced
Logical Libraries, graphical view	Not available with this interface	Library > Logical Libraries > Actions > Graphical View
Logical Libraries, Manage (Basic Mode)	Not available with this interface	Library > Logical Libraries > Actions > Manage Logical Library (Basic Mode)
Logical Libraries, Manage (Expert Mode)	Not available with this interface	Library > Logical Libraries > Actions > Manage Logical Library (Expert Mode)
Logical Libraries, Mode configure	Not available with this interface	Library > Logical Libraries > Actions > Manage Logical Library (Basic Mode) or Manage Logical Library (Expert Mode) See Random and Sequential Logical Library modes .
Logical Libraries, Mode status	Not available with this interface	Library > Logical Libraries
Magazines, open	See Accessing cartridges .	Library > Modules and Magazine > Actions > Unlock Magazine
Network settings	Configuration > Network Settings	Settings > Network > Ethernet
Notifications, configure	Not available with this interface	Settings > Notifications
Operator Panel, session lock timeout	Not available with this interface	Settings > Security > GUI
Password Policy	Not available with this interface	Access > Local Password Policies
Path Failover license key, add or delete	Not available with this interface	Settings > Library > Licensed Features
Port Settings, modify	Not available with this interface	Drives > Actions > Modify Port Settings
Remote Logging (rsyslog), configure	Not available with this interface	Settings > Notifications > Remote Logging (rsyslog)
Reset, rediscover devices	Not available with this interface	Settings > Library > Advanced
Reset Internal IP Range	Configuration > Reset > Reset Internal IP Range	Not available with this interface
Reset Library	Not available with this interface	Library > Actions > Reset Library
Reset Drive	Not available with this interface	Drives > Actions > Reset Drive
Sequential mode, configure	Not available with this interface	Library > Logical Libraries > Actions > Manage Logical Library (Basic Mode) or Manage Logical Library (Expert Mode) See Random and Sequential Logical Library modes .
SNMP, configure	Not available with this interface	Settings > Notifications > SNMP
SNTP (Simple Network Time Protocol) Synchronization, configure	Not available with this interface	Settings > Library > Date and Time > SNTP (Simple Network Time Protocol) Synchronization
SSL, enable or disable	Not available with this interface	Settings > Security > GUI > Secure Communication
Session Timeout	Not available with this Interface	Settings > Security > GUI > Session Timeout
Time Zone, set	Not available with this interface	Settings > Library > Date and Time > Time Zone
Unlabeled Media, allow	Not available with this interface	Settings > Library > Advanced
Users, Access recovery	Configuration > User Accounts	Not available with this interface
Users, Modify User Passwords	Not available with this interface	Access > Local Users > Actions > Modify User Passwords
Users, Modify Role Permissions	Not available with this interface	Access > Local Users > Actions > Modify Role Permissions
Users, Modify Operator Panel PIN	Configuration > User Accounts	Access > Local Users > Actions > Modify Operator Panel PIN
Users, Add	Not available with this interface	Access > Local Users > Add User For information about different roles, see Managing .
Users, Remove	Not available with this interface	Access > Local Users > Actions > Remove User

- [Default settings](#)
The library is set to default settings when first purchased. Many of these settings can be customized.
- [Methods of cleaning drives](#)
Automatic cleaning of tape drives is disabled by default in the library. However, automatic cleaning of tape drives is recommended for this library. It is also possible to initiate manual or host cleaning methods.
- [Accessing cartridges](#)
Each magazine has a button that provides an easy way to open a magazine.
- [Configuring Library Managed Encryption](#)
Library-Managed Encryption (LME) is a built-in feature that is enabled by using a purchased license.

Default settings

The library is set to default settings when first purchased. Many of these settings can be customized.

Table 1. Default settings

Parameter	Default	Configuration Reset Default Settings
User Accounts		

Parameter	Default	Configuration Reset Default Settings
Administrator login	User = administrator Management GUI Password = adm001	NOT reset
Local user accounts	Local Default Users = administrator Number of Custom User = 0	x
User Account Settings		
Password rules	Min. number of characters 8 Min. number of uppercase alphabetic characters 1 Min. number of lowercase alphabetic characters 1 Min. number of numeric characters 1 Min. number of special characters 0 Max. number of identical consecutive characters 2 Max. number of failed logins 3 Max. number of days before PW must be changed 90 Number of PW changes before it can be used again 3	x
Management GUI Restricted Login	Disabled	x
Allow I/O Station/Magazine access by monitor role	Disabled	x
Session Locking	Disabled	x
Remote Authentication (LDAP) Configuration	Disabled	NOT reset
Network configuration (eth0)		
Host name	Blank	NOT reset
IP address	(dhcp)	NOT reset
Subnet mask	(dhcp)	NOT reset
Default gateway	(dhcp)	NOT reset
Auto Negotiate	Enabled	NOT reset
Speed	Auto	NOT reset
IPv4	Enabled	NOT reset
DHCPv4	Enabled	NOT reset
IPv6		NOT reset
IPv6 Prefix	Enabled	NOT reset
Static v6	Disabled	NOT reset
IPv6 Method	Disabled	NOT reset
DHCPv6	Disabled	NOT reset
DNS1 and DNS2 Configuration for IPv4	(dhcp)	NOT reset
DNS1 and DNS2 Configuration for IPv6	Disabled	NOT reset
Network Access Services		
Primary Network Interface (eth0)	Enabled	NOT reset
Secondary Network Interface (eth1)	Disabled	NOT reset
HTTPS	Disabled	NOT reset
Self Signed SSL Certificate	No file	NOT reset
Internal IP (eth2)		
Internal network IP	IP Range defined with Operator Panel	NOT reset
I/O Station/Magazine		
I/O Station	Enabled	x
I/O Station/Magazines Allow Access by monitor role	Disabled	x
Logical Libraries	Disabled	All Logical libraries are deleted
NTP/SNTP Setting	Disabled	NOT reset
Date	Blank or existing	NOT reset
Time	Blank or existing	NOT reset
Time Zone	GMT	NOT reset
Sequential Mode	Disabled	x
Email Notifications (SMTP)	Disabled	x
SNMP		
SNMP v1, v2	Disabled	x
Licensed Features (need license key for enablement)		
Encryption	Disabled	NOT reset
Path Failover	Disabled	NOT reset
SCSI Defaults		
Product Name - Marketing Name	TS4300	x
Library Product ID - INQUIRY Product ID String	3573-TL	x
Library Vendor ID - INQUIRY Vendor ID String	IBM	x
SCSI element addressing	Starting element addresses: Drives = 1 I/E Elements = 101 Storage Slots = 1001	x
Miscellaneous settings		
Barcode format returned to host	Align left	x
Barcode length returned to host	Eight leftmost characters	x
Language settings	English	NOT reset
Auto Clean	Disabled	x

Parameter	Default	Configuration Reset Default Settings
Media Barcode Compatibility Check	Enabled	x
Empty Slot/Unlabeled Cartridge Detection	Enabled	x
Management GUI Timeout	30 minutes	x
Drive Defaults		
Drive speed and topology settings	Automatic/Automatic	x
Odometer	Enabled	NOT reset

Methods of cleaning drives

Automatic cleaning of tape drives is disabled by default in the library. However, automatic cleaning of tape drives is recommended for this library. It is also possible to initiate manual or host cleaning methods.

The head of every tape drive in the tape library must be kept clean to prevent errors that are caused by contamination. To help you keep the drives clean, IBM provides a cleaning cartridge with the library. The library uses the cleaning cartridge to clean the drive with whatever cleaning method that you choose. In all methods, cleaning is done after the data cartridge is unloaded from the drive and before the next load.

If you put the cleaning cartridge into a storage slot that is assigned to a logical library, it is visible to any hosts that are associated with that logical library. If you don't want the cleaning cartridge visible to any hosts, put it into a storage slot that is not assigned to any logical libraries.

It is the operator's responsibility to monitor cleaning cartridge usage and replace cleaning cartridges as necessary. This tape library provides multiple ways to monitor and manage cleaning cartridges. If SNMP traps are enabled, a trap is generated when a cleaning cartridge expires. It is also possible to use the Management GUI to monitor the cleaning cycles that remain on a cleaning cartridge.

Three methods of cleaning are available.

Automatic cleaning

Automatic cleaning enables the library to automatically respond to any tape drive's request for cleaning and to begin the cleaning process. The cleaning process is transparent to any host application that uses the library.

Select Auto Clean to enable the auto cleaning feature. When enabled, the library automatically initiates a cleaning operation when media is unloaded from a drive that requires cleaning instead of creating a warning event when a drive requires cleaning. For reliable operation, enable Auto Clean for each logical library and ensure that the library has a valid cleaning cartridge.

It is preferred to put the cleaning cartridge into a storage slot that is not assigned to a logical library. If you put the cleaning cartridge into a storage slot that is assigned to a logical library, it is visible to any hosts that are associated with that logical library. If you don't want the cleaning cartridge to be visible to any hosts, put it into a storage slot that is not assigned to any logical libraries.

When a cleaning operation is initiated, the library first attempts to use an unexpired cleaning cartridge from the same logical library as the tape drive. If the logical library does not contain an unexpired cleaning cartridge, the library attempts to use an unexpired cleaning cartridge from a storage slot that is not assigned to a logical library. The library does not use a cleaning cartridge from a different logical library. When auto cleaning is enabled, ensure that each logical library has an unexpired cleaning cartridge. Or, place at least one unexpired cleaning cartridge into a storage slot that is not assigned to a logical library.

After the initial configuration, Auto Clean can be turned on or off by accessing the Logical Library Wizard Expert Mode.

1. Go to Library > Logical Libraries > Actions > Manage Logical Library (Expert Mode).
2. Select the logical library, then click Edit.
3. Click Next to go to the General Settings screen.
4. Check or clear Auto Clean, then click Next on the subsequent screens.
5. If a change was made, click Finish and the Logical Libraries are reconfigured. If no changes were made, click Cancel.

Note: IBM recommends enabling the Auto Clean function on the library. With the Auto Clean function enabled, drive cleaning occurs automatically. The only time Auto Cleaning must be disabled is when your backup application requires that it has control.

Manual cleaning

Manual cleaning requires that you select a menu option from the Management GUI to clean one or more of the tape drives. Manual cleaning is always supported.

Host cleaning

Host cleaning enables the backup application to define and control the cleaning process. Automatic and manual cleaning use the CLNxxx VOLSER. It is mandatory to put the cleaning cartridge into a storage slot that is assigned to a logical library. The host needs to "recognize" that the cleaning cartridge is available for use.

Note: For tape cartridge information, see [Supported tape cartridges](#).

Accessing cartridges

Each magazine has a button that provides an easy way to open a magazine.

See [Front panel](#).

Each module can be configured to have a portion of the right magazine that is designated as an I/O station or this option can be disabled.

- To open the I/O station, press the magazine button for less than 3 seconds.
- To open the entire magazine, press the magazine button for more than 3 seconds.

Notes:

1. If a magazine is opened, no other magazines or I/O stations can be opened.
2. If during the magazine open process the magazine is not opened within 30 seconds, the magazine locks.
3. The user must pull out the magazine, as the magazine does not eject.

The LED also provides an indicator of the current state of that magazine.

Table 1. Magazine state

Magazine state	LED state	Description
Closed	Steady ON	I/O station is enabled.
Closed	Slow Flash	Magazine open is in process.
Closed	Fast Flash	Magazine is opened.
Closed	OFF	I/O station is not enabled.
Opened	OFF	Magazine is opened.

Configuring Library Managed Encryption

Library-Managed Encryption (LME) is a built-in feature that is enabled by using a purchased license.

The LME feature can be ordered from the factory, or you can order it as a field upgrade. To order a feature, contact your IBM® Sales Representative or Business Partner. See [Optional Features](#).

Two versions of Library-Managed Encryption are available for configuration.

- [Key Management Interoperability Protocol \(KMIP\) Encryption \(v1.2\)](#)
- [Security Key Lifecycle Manager \(SKLM\) for z/OS® Encryption](#)

Access the wizard from the Actions menu with the Manage Encryption option.

Notes: Before you run the Encryption wizard.

- Confirm that the Library-Managed Encryption license is activated on the Settings > Library > Licensed Features page.
- Verify that the server is available on the network and is configured for use with this library. For information on configuring servers for use with the library, see the server documentation.

Note: If you plan to use the IBM Security Key Lifecycle Manager (SKLM), go to [Related publications](#) for information on setup and configuration.

- If Library Encryption settings are cleared and reconfigured, you're required to accept the new certificate on the server when the Library Self-Signed Certificate is used.

Key Management Interoperability Protocol (KMIP) Encryption

1. In the Actions menu, click Manage KMIP Encryption to start the wizard.
2. The Logical Library Selection screen displays the KMIP configuration options that can be set as the default for all logical libraries, or on a per logical library basis. The second section provides the option to copy the KMIP configuration settings to all logical libraries (default) or to specified logical libraries.
3. The Wizard Information screen displays information about the wizard. On this screen, it's also possible to Reset Encryption Settings. If the library configuration is complete and the KMIP server is available on the network, click Next.
4. The Certificate Option screen displays the different certificate options that can be used to establish a secure communication to the KMIP server. You can select from the following options:
 - Library Self-Signed Certificate (default option) - A self-signed certificate that is generated by the library is used.
 - Uploaded Certificate - Upload a PKCS #12 file that includes a certificate and corresponding key.
 - Generate Certificate Request (CSR) - A CSR is generated by the library that must be signed by a CA server. This method requires a CA certificate that must be provided during the wizard steps.
 - Certification Configuration
 - Library Self-Signed Certificate – skip to the next step.
 - Uploaded Certificate
 - Upload the PKCS #12 file in the certificate area on the Certificate Option screen.
 - If this file requires a password, it must be provided in the Certificate Password input field. If no password, the field can be left empty.
 - After successfully upload of the certificate, click Next.
 - Generate Certificate Request (CSR)
 - The Certificate Authority Information screen displays prerequisites for using the KMIP certificate. When the prerequisites are met, click Next.
 - The Certificate Authority Certificate Entry screen displays instructions for obtaining the CA certificate for the KMIP server. Follow the instructions to copy the CA certificate from the Management Console. Paste the CA certificate into the wizard and then click Next.
 - The Library Certificate Information screen displays information about the next wizard steps. Click Next.
 - The KMIP Client Configuration screen provides options for two types of server authentication.
 - If your KMIP server uses a client username and password for authentication, enter the username and password that were specified on the KMIP Management Console for the library.
 - If your KMIP server uses certificate validation for authentication, select Enable KMIP Certificate only authentication. Select this option if you use a KMIP server that doesn't support a client username and password. This default method is used when KMIP is used with the IBM Security Key Lifecycle Manager.
 - In the KMIP Server Configuration screen, enter the IP address or fully qualified hostname and port number for up to ten KMIP servers. Also, choose which key server type services the encryption keys. You can select from the following options:
 - IBM SKLM - IBM Security Lifecycle Manager 2.6.0 or higher KMIP server.
 - KMIP Compatible - Key server that is supporting the OASIS standard key management interoperability protocol (KMIP).
 - To verify access to the KMIP servers, click Connectivity Check.
 - Check at the KMIP server side that the server accepts the certificate of the library.
 - The Setup Summary screen displays the settings that are collected by the wizard. Verify that the settings are correct and that no errors are in the Done column.
 - If you need to modify any settings or fix any issues, either click Back to reach the applicable screen or Cancel to leave the wizard to fix the issues and return later.
 - If the settings are correct and no errors are reported, click Finish.

When the wizard finishes, the Library Managed Encryption (KMIP) encryption mode is selectable in the Logical Library Wizard (Expert Mode) on the Library > Logical Libraries page.

Security Key Lifecycle Manager (SKLM) for z/OS Encryption

1. Go to the Library menu. Then, go to Logical Libraries. Select Actions, then select Manage SKLM for z/OS Encryption.
2. Enter the IP address and the port of the SKLM z/OS server, then click Modify.
3. Go back to Actions and select Manage Logical Library (Expert Mode).
4. On the Expert Logical Library Wizard screen, click General Settings.
5. Next to Encryption Mode, choose Library Managed Encryption (SKLM for z/OS) (Licensed).
6. Click Next, and then click Finish Configuration.
7. A message appears when the Logical Library was successfully enabled for SKLM for z/OS.
8. Go to Settings > Security > Encryption. The Security Encryption Status and the Logical Library Encryption Status shows Library Managed Encryption (SKLM for z/OS) as Enabled.

Key Path Diagnostics

The Key Path Diagnostic test checks all communication paths to ensure that a key can be transmitted from the encryption key servers to the drive to properly encrypt and decrypt the tape cartridges.

The test consists of two parts. The first part, the drive test, verifies whether the communication between library and drive is working properly. This test is run only on the drives that are configured to library-managed encryption (LME).

The second part verifies the communication between the library and the encryption key servers. If the secondary ethernet port is enabled and configured, the tests are run on both ports separately.

The test consists of four subtests:

- Ping

This test checks if the key server can be reached. If ICMP requests are blocked on the server side, this test fails as well. Therefore, the following tests are run regardless the result of the ping test.

- SSL/TLS

This test tries to establish a SSL/TLS connection with the key server. If this test fails, the following tests are skipped because they would also fail. This test is skipped if SSL/TLS is not enabled.

- Key Server Login

This test is run only in combination with a KMIP encryption server since SKLM currently does not support login. If this test fails, the following Key Retrieval test is skipped because it would also fail.

- Key Retrieval

This test requests a key from the encryption server. For SKLM servers, a key from the key pool is requested. On other servers, the library acquires a specific diagnostic key.

Troubleshooting

Use the information in this section to troubleshoot any issues with your library setup and configuration.

Attention: This library is designed to operate when installed in a rack with the rack rail kit or on a tabletop. Operating the library without installing it correctly in the rails might cause errors. Placing any weight on top of the library might also cause errors. Expanded library configurations on tabletops are not supported.

- [How the library reports problems](#)
The library uses advanced problem detection, reporting, and notification technology to alert customers of problems as soon as they occur.
- [Identifying a failed component](#)
Follow the procedures to identify a failing component.
- [Running library tests](#)
The library provides tests to verify library operations.
- [Troubleshooting Guide](#)
Refer to this table of symptoms or errors that might occur with the tape library and the installed tape drives.
- [Pre-call checklist](#)
If you have questions or problems with the library, complete these steps before a call to IBM technical support is placed.
- [Contacting IBM technical support](#)
Follow the procedures to contact IBM technical support.
- [Diagnostic information](#)
This section provides various diagnostic tools and information.

How the library reports problems

The library uses advanced problem detection, reporting, and notification technology to alert customers of problems as soon as they occur.

It completes numerous self-tests to monitor the library's temperature, voltage and currents, and standard library operations. These tests monitor the library when the library is powered ON, and during normal operation when the library is idle.

If the test detects a problem, the library generates a message that identifies which component is likely causing the problem. The library's Error LED and Attention LED might turn ON to indicate an abnormal state. If the problem is not severe, the Attention LED turns ON and the library continues to operate normally. If the problem is not recoverable, the Error LED turns on and an error message is displayed on the Operator Panel.

When the library generates an attention event or an error event, support staff can be notified immediately by setting up email event notification or SNMP trap notification. The type of event that generates email notification or SNMP trap notification can be selected to limit the number of events to a specific priority level.

Customers can frequently resolve a simple problem themselves with the information that is available in [Troubleshooting guide](#). If you do not find the error listed in [Troubleshooting guide](#), refer to [Pre-call checklist](#).

Identifying a failed component

Follow the procedures to identify a failing component.

1. Management GUI: Activate the UID LEDs from the Library > Actions > Turn Identifier Light On or Off screen. This action illuminates the blue LED on the front and rear of the Base Module to identify the library that contains the failed module or component.
2. Identify the module within the library that contains the failed component.
 - a. In the upper left of the Home screen, locate the module that indicates an error.
 - b. Click or tap the module for information on the failed component.

Running library tests

The library provides tests to verify library operations.

- On the Management GUI, go to Library > Actions > Tests and choose the library test that you want to run.
 - Library Verify
 - Demo Mode
 - Drive Test
 - Slot to Slot Exerciser
- On the Operator Panel, go to Maintenance > Library Tests to run tests from the Operator Panel.
 - Demo Mode
 - Drive Diagnostics

Troubleshooting Guide

Refer to this table of symptoms or errors that might occur with the tape library and the installed tape drives.

The table provides actions to correct the problems. If replacement parts are needed, go to [Replacement parts](#). See [Contacting IBM technical support](#).

Table 1. Resolving errors

Problem	Solution
Event code/Attention information on Management GUI or Library	
Event code that is shown on Event Ticket on the Management GUI. See How the library reports problems .	• Look up the error code. See Event Codes. • Try to resolve the failure. • If necessary, power cycle the library.
Failure/Attention Indication on Operator Panel display.	Review tickets on the Check Event Ticket Log on Management GUI.
Attention LED is lit on the front or the rear of the Base Module.	Review tickets on the Check Event Ticket Log on Management GUI.
Failure/Attention Indication on Management GUI Library Dashboard	Tap the icon to see information about the event.
Attention LED and Cleaning LED are lit.	This problem is likely caused by a drive that requires cleaning. • Check Event Ticket Log on Management GUI.
Single Character Display (SCD) is shown on drive.	1. Review tickets on the Check Event Ticket Log on Management GUI. 2. Use SCD. See Drive Single Character Display.
The Attention LED is lit but the Cleaning LED isn't lit after a cartridge load.	The library was unable to complete the requested operation with the selected tape cartridge. • Use cartridges that are compatible with the drive type. • Use the correct type of cartridges for the operation. For example, use a cleaning cartridge for cleaning. • Make sure that you're using a Universal cleaning cartridge.
The Cleaning LED is lit after a cleaning cartridge was used.	The cleaning cartridge is expired. (A cleaning cartridge expires after 50 cleaning cycles). • Replace the cleaning cartridge.

Problem	Solution
A particular cartridge sets off the Attention LED and possibly the Cleaning LED.	Retry the operation with a different cartridge. If the Attention LED is cleared, and then immediately redisplay each time that a particular cartridge is reloaded, that cartridge must be suspected as defective. - Export the cartridge and load a known good cartridge. In some cases, a cartridge can be worn out, the memory is defective, or was formatted as a Firmware Upgrade cartridge. - Any cartridge that is suspected of being defective or contaminated must NOT be reused in any drive. - If the bad cartridge is a cleaning cartridge, it might be expired.
Event Notification on Host, SNMP, or Email	
Host receives error message.	- Use ITDT. See IBM Tape Diagnostic tool (ITDT). - Use Sense Data. See Sense data.
SNMP Monitoring system receives trap.	Check Event Ticket Log on Management GUI.
Event is received by email notification.	Check Event Ticket Log on Management GUI.
Cartridge Movement Problems	
Cartridge failing to load and unload properly	Note: - The tape drive must rewind the tape before it is ejected. The amount of time for this procedure can vary, depending on how the tape was used. See Supported tape drives. - The tape drive performs media optimization Media optimization on the first load of a cartridge. The amount of time for this procedure can vary. See Supported tape drives. Follow these instructions for removing tape from the tape drive: - Stop all host activity. - Check drive status by using any of the following methods: SCD display. ITDT - Indicator lights on a library if the drive is installed in a library - If the drive is in the middle of performing an operation, wait until the drive is idle before attempting any further steps. - Attempt to unload the cartridge. - Power down the drive. - Disconnect the host cable from the drive. - Power on the drive, and wait until the tape drive is idle or ready. - Attempt to unload the cartridge. Follow these steps to inspect a cartridge for damage: - Check that the leader pin is attached and properly seated by opening the cartridge door and observing the pin's placement. See Repositioning a leader pin. - Inspect the cartridge case, the cartridge door, and the write-protect switch for damage. - Inspect the rear of the cartridge (the part that you load first into the tape load compartment) and ensure that there are no gaps in the seam of the cartridge case. See 1 in Figure 1 and 4 in Figure 2. If there are gaps, the leader pin might be dislodged. Go to Repositioning or reattaching a leader pin. - Try loading or unloading another tape cartridge. - If it fails, contact your service representative for more problem determination. - If it is successful, discard the cartridge that originally failed. Notes: - If a damaged or mishandled cartridge is the problem, see Cartridge care and handling for instructions about handling the media. It is possible that your other cartridges might also be damaged. - If your cartridge does not eject properly, contact your service representative.
Tape is stuck in drive.	Try the following steps, in this order, to remove the stuck tape. Note: The tape drive must rewind the tape before it's ejected. This procedure can take up to ten minutes or more, depending on how much tape must be rewound. When the tape is rewound, the eject cycle takes fewer than 16 seconds. The Ready light flashes while the tape rewinds. Wait for the tape to finish rewinding before another operation is attempted. - Stop all Host Activity. - Attempt to unload or move the cartridge to a slot. - Power down the library, disconnect the cable from the drive, power on the library, and wait until the tape drive is idle or ready. Attempt to Move Cartridge to Slot. - Attempt a Eject Cartridge from Drive as an emergency unload operation. Important: Inspect the tape cartridge that was stuck. Damage or misplaced labels on the cartridge might cause the load/unload failure. Discard any tape cartridge that is found to have issues.
Cartridge can't be removed from storage slot.	- Unlock the magazine and extend it to access the storage slot. - Grasp the cartridge and remove it from the storage slot. Some tapes need to be inserted and removed several times to condition them for free movement in and out of the magazine. - Check the bar code label and verify that it's secure to the cartridge. - Check the cartridge for damage.
Other Library Problems	

Problem	Solution
Device doesn't power on.	- Check all power cord connections. - Check the LEDs on the power supplies. - Make sure that Power on the front panel was pressed, and the green Ready LED is lit. - Make sure that the outlet has power. Try another working outlet. - Replace the power cord.
No message appears on the Operator Panel display.	- Check all power cord connections. - Check the LEDs on the power supplies. - Make sure that Power on the front panel was pressed, and the green Ready LED is lit. - Make sure that the outlet has power. Try another working outlet.
Can't load the cleaning cartridge.	- Make sure that you're using an LTO cleaning cartridge. - Make sure that the cleaning cartridge isn't expired. A cleaning cartridge expires after 50 cleaning cycles.
User account locked.	From the Configuration > User Accounts > Access Recovery page, you can receive a temporary administrator password for login to the Management GUI that is valid for two hours.
Tape Drive or Media Problems	
Can't write to or read from tape.	- Make sure that the cartridge isn't a WORM cartridge that was already used. - Make sure that the cartridge is write enabled (move the write-protect switch to the enabled position). - Make sure that the data cartridge is compatible with the drive model. See Supported tape cartridges. - Make sure that you're using an LTO cartridge that isn't degaussed. CAUTION: Don't degauss LTO cartridges! - Make sure that the cartridge wasn't exposed to harsh environmental or electrical conditions and isn't physically damaged in any way. - Many backup applications don't read or write to cartridges that were created with a different backup application. In this case, you might need to run an erase, format, or label operation on the cartridge. - Make sure that you understand any data protection or overwrite protection schemes that your backup application might be using, which might prevent you from writing to a specific cartridge. - Retry the operation with a different, known good tape. - Clean the tape drive.
A cartridge that is recently imported from a different environment is causing issues.	Media that is moved from one environment to another can cause issues until it acclimates to the new conditions. A cartridge must be acclimated for at least 24 hours before it's used, particularly if it was stored at a substantially different temperature or level of humidity than the device.
The library displays incorrect bar codes.	- Verify that the label is properly applied. - Verify that the label isn't soiled.
Cleaning or data cartridge incompatible with drive.	- Check the event log to see which cartridge is incompatible. - Make sure that you're using data and cleaning cartridges that are compatible with the drive and model of your device. - Make sure that you're using the correct cartridge type for the operation. The device automatically unloads incompatible cartridges, and the Attention LED flashes. - Export the media.
Connection Problems	
Fibre Channel connection problems	Check Drive Status screen to check the link connection for your tape drive. - For each available port, run a drive Wrap Test. - Unplug the Fibre Channel cable, then plug in the Fibre Channel wrap tool. - Go to Library > Actions > Tests > Drive test and run the Fibre Channel wrap test. - If the drive test fails, follow the Pre-call checklist. - If the FC Wrap Test passes, continue with the next steps. - Check that the Fibre Channel speed is set to either match the HBA/switch speed or set to Automatic. - Verify that cables aren't damaged. - Verify that cables are securely connected on both ends. - Verify Host Connectivity. - Use ITDT to debug the problem.
SAS connection problems	Check Drive Status screen to check the connection for your tape drive. - For each available port, run a drive Wrap Test. - Unplug the SAS cable, then plug in the SAS wrap tool. - Go to Library > Actions > Tests > Drive test and run the SAS wrap test. - If the drive test fails, follow the Pre-call checklist. - If the SAS Wrap Test passes, continue with the next steps. - Verify that cables aren't damaged. - Verify that cables are securely connected on both ends. - Verify Host Connectivity. - Use ITDT to debug the problem.

Problem	Solution
Can't connect to the Management GUI.	- Verify that the Ethernet cable is connected to the Base Module's controller card and to the LAN. - Verify that the link LED on the RJ45 (LAN) connector is lit when the device is turned on. If the LED isn't lit, the device isn't communicating with the LAN. See your network administrator for help. - Verify that the device is configured with a valid static network address or DHCP is enabled so the device can obtain a network address. If DHCP is used, write down the device's network address from the Operator Panel login screen. If a valid DHCP address isn't available, the library isn't communicating with the DHCP server. See your network administrator for help. - Enter the library's IP address into the address bar of a web browser that is connected to the same LAN as the device. If the Management GUI page doesn't display, ping the device's IP address. If the ping fails, check that no firewalls or other obstructions to network traffic exist between the computer with the web browser and the device. See your network administrator for help.
Can't connect to Key Management Server for LME.	- Run the Encryption Connectivity Check or Key Path Diagnostics and review the results message. See Locating Management functions. - Validate that the server configuration properties file includes support for TLS 1.2. See the encryption server documentation. - If SKLM v2.7 or higher is used and you're using the Library Self-Signed Certificate. For one time only, you might need to reset your encryption settings to clear out old versions of the self-signed certificate, reconfigure encryption, then accept the new self-signed certificate on the encryption server. See Configuring Library Managed Encryption. - Check that your certificate algorithm is supported by your version of the server.
Host application reporting SCSI timeout	Note: - The tape drive must rewind the tape before it is ejected. The amount of time for this procedure can vary, depending on how the tape was used. See Supported tape drives. - The tape drive performs media optimization on the first load of a cartridge. The amount of time for this procedure can vary. See Supported tape drives. The procedure for SCSI timeouts varies depending on whether timeout is consistent or intermittent, and on your drive configuration. Follow these steps to troubleshoot a SCSI timeout: 1. Stop all Host Activity. 2. Check drive status by using any of the following methods: • SCD display • ITDT • Indicator lights on a library if the drive is installed in a library 3. If the drive is in the middle of performing an operation, wait until the drive is idle before attempting any further steps. 4. Validate that the drive has the latest firmware. 5. Check with software application provider for any updates. 6. Check whether the tape drive power is on. 7. Power down the drive. 8. Verify that the SAS or FC cable is connected properly to the server and to the tape drive. 9. Replace SAS or FC cable if it shows any signs of damage. 10. Power on the drive and wait until the tape drive is idle or ready.

Pre-call checklist

If you have questions or problems with the library, complete these steps before a call to IBM technical support is placed.

Note: Where instructions refer you to the web, go to <http://www.ibm.com/storage/support/lto>.

1. Verify that you exhausted all troubleshooting options. See [Troubleshooting Guide](#).
2. Collect library and drive logs. See [Locating Management functions](#).
3. Verify that the library and drive firmware is at the most recent level. See [Locating Management functions](#).
4. Verify that your device drivers are at the most recent level.
 - For the current release of IBM® device drivers, see [Supported Device Drivers](#).
 - For the current release of device drivers by independent software vendors (ISVs), go to the appropriate third-party website.
5. Verify whether your hardware and software configuration is supported. See [Host requirements](#).
6. Check the hardware and connections:
 - Ensure that the host interface cable connector does not contain bent or recessed pins.
 - Ensure that all retention screws for the host interface cable and terminator are securely tightened.
 - Verify the host connection. See [Verifying the host connection](#).

If you still have a problem after these steps are completed, see [Contacting IBM technical support](#).

Contacting IBM technical support

Follow the procedures to contact IBM technical support.

Important: This tape library is a customer installed unit. The customer is responsible for the setup and maintenance of the tape library. The customer is charged for service if a service contract is not in place.

- Complete the steps in [Pre-call checklist](#) before a call is placed to IBM® technical support.
- Before IBM technical support is called, the customer is responsible for following published LTO diagnostic procedures, including any needed update to the current level of firmware.
- The IBM Support Center assists with problem determination and can initiate shipment of a replacement part, if needed, to the customer's location. Transportation costs, both ways, are paid by IBM. The replacement part becomes the property of the customer in exchange for the failed part, which becomes the property of IBM.

The customer is responsible for packing the failed part into the shipping carton that contained the replacement part. Failure to return the failed part to IBM within 30 days results in a bill sent to the customer for the new list price. The customer is responsible for installing and setting up all replacement parts.

- Failure to use the carton in which the replacement part was received, or failure to properly pack the returned part, can result in charges that are incurred for damage to the failed part during shipment.
- Before you call technical support, follow these steps that help you take full advantage of your call.
 1. Be prepared to provide
 - Machine type and Model name
 - Serial number of the library
 - Hardware configuration, including firmware versions, drive types, modules
 - Type of host, operating system version, device driver information, software application, and version, clock speed, RAM, network type, network version
 - A brief description of the problem, including Event Ticket information. See [How the library reports problems](#).
 2. Review all documentation carefully. (Experience demonstrates that most questions are answered in your documentation).
 3. Be prepared to explain whether the software or hardware worked properly anytime in the past. Have you changed anything recently?
 4. Pinpoint the exact location of your problem, if possible. Note the steps that led to the problem. Can you duplicate the problem or is it a one-time occurrence?
 5. Note any host error messages displayed. Write down the exact error message.
 6. If possible, call while at your computer, with the library installed and turned on.
- The IBM Support Center assists with problem determination and initiates shipment of a replacement part, if needed, to your location. To contact IBM technical support:
 - In the US: 1-800-IBM_SERV (1-800-426-7378).
 - All other Countries or Regions: <http://www.ibm.com/planetwide/>.
 - Select your country, then under Technical Support, click Open Service Request.

Diagnostic information

This section provides various diagnostic tools and information.

- **IBM Tape Diagnostic tool (ITDT)**
The IBM Tape Diagnostic Tool (ITDT) is an independent tool that provides diagnostics on tape drives and libraries. ITDT has multiple functional capabilities and is a quick, convenient, and efficient method for drive firmware updates. As a note, drive memory dump retrievals are completed by the tool as well.
- **Event codes**
Events are used in the library Ticket and Event system to store all types of events with a unique event code and event description.
- **TapeAlert flags**
This section is intended to provide information to the reader about the tape drive by using TapeAlert technology.
- **Sense data**
When a drive encounters an error, it provides sense data as a response to the host.
- **Drive Error Codes: Single-character display (SCD)**
SCD drive error codes give descriptions of the errors and messages that pertain to the drive.

IBM Tape Diagnostic tool (ITDT)

The IBM Tape Diagnostic Tool (ITDT) is an independent tool that provides diagnostics on tape drives and libraries. ITDT has multiple functional capabilities and is a quick, convenient, and efficient method for drive firmware updates. As a note, drive memory dump retrievals are completed by the tool as well.

The IBM® Tape Diagnostic tool (ITDT):

- Runs quick or extended diagnostic tests on tape drives. If the library is online to the server/host where the tool resides, ITDT communicates with the drive through the library to load and unload a test cartridge, exercising some library functions.
- Retrieves firmware memory dumps from tape drives and libraries.
- Completes a firmware update on tape drives or libraries. See note about library firmware update.
- Tests the performance of the environment by completely writing a cartridge and measuring performance.
- Retrieves and displays cartridge information.
- Verifies the encryption environment.
- Does not require special device drivers.
- Is available for most major platforms.
- Scans the host interface and finds and displays for selection all IBM LTO devices. The tool does not display non-IBM devices.

The IBM Tape Diagnostic tool (ITDT) is available as a command line utility and a graphical user interface (GUI) version.

- The IBM Tape Diagnostic Tool (ITDT) is started by entering the executable file from the directory where the tool is located. The Help feature gives a brief explanation of each function and shows the syntax.

Note: Be sure that you have the most current version of ITDT if you are updating firmware on a recent drive type. Before ITDT is used, verify that your library host operating system is at the current released level. This action ensures optimum read/write operations for diagnostic tests.

IBM maintains the current levels of ITDT on the web. Go to <http://www.ibm.com/support/fixcentral> and follow these steps to access this material.

1. From the Fix Central web page, choose the Select Product tab, select System Storage from the Product Group list.
2. Select Tape Systems from the System Storage list.
3. Select Tape drivers and software from the Tape systems list.
4. Select IBM Tape Diagnostic Tool ITDT from the Tape drivers and software list.
5. Select the appropriate operating system from the Platform list and click Continue.
6. Select the appropriate version from the list.

Additional information about the IBM Tape Diagnostic Tool (ITDT) is included in the *IBM® Tape Device Drivers Installation and User's Guide*, and can be found on the web at <https://www.ibm.com/docs/en/ts4300-tape-library?topic=guide-tape-diagnostic-tool-itdt>.

Event codes

Events are used in the library Ticket and Event system to store all types of events with a unique event code and event description.

Event Reporting System

These event codes are shown on the user interfaces as the resulting code for any type of event at the highest level of information. No internal error code is shown at this level.

Events are sent by the library to different recipients like SNMP targets or email notification. These events have a common structure and unique codes for every type of event.

The event code system is used for the following events:

- Error event (2XXX)
- Warning event (4XXX)
- Configuration event (8XXX)
- Informational event (9XXX)

Event code structure

Example event code

Event: 2057 - Robotics shipping lock in incorrect position

The event log with the library also includes a date and time stamp for each event. Press the associated time stamp to see the event code and a description of the event. The date and time format can be changed in the Date/Time Format section in the Management GUI.

- **mm.dd.yyyy**
- **dd.mm.yyyy**
- **yyyy.mm.dd**

The time format can be set for 12 hours or 24 hours.

- 12 hour: **hh.mm.ss am/pm**
- 24 hour: **hh.mm.ss**

Where

- **yyyy** is the year.
- **mm** is the month.
- **dd** is the current day.
- **hh** is hours.
- **mm** is minutes.
- **ss** is seconds.

Resolving an error code

1. Record the error information that is displayed on the Operator Panel display or Management GUI screen.
2. If possible, cycle library power and retry the operation.
 - If the error does not recur, run Library Verify before normal library operation is continued.
3. If the error recurs, click the event to see its details. If available, click Troubleshooting on the Event Ticket Details screen to get suggestions on how to fix the error. Click OK to close the Event Ticket Details screen.
4. When the proposed solution is applied, run Library Verify before normal library operation is continued.

Complete the steps in [Resolving an error code](#) before you complete the User Action that is listed in the various Event Codes.

- [Main error events](#)
- [Warning error events](#)
- [Configuration Change events](#)
- [Informational events](#)

Main error events

Table 1. Main error events

Event Code	Message Text and Description	Details and Solution
2000	Move Cartridge failed.	Verify the source and destination elements and retry the move operation.
2002	The initial module discovery (detection of expansion modules) failed.	Ensure that all modules are powered and have the interconnection cables properly attached. Also, ensure that the module alignment locks (at the rear of module) are in the correct positions.
2003	The library's temperature exceeded the critical temperature threshold.	Check to ensure that • The drive cover plates are installed where no drive exists. • All power supplies are installed. • The ambient room temperature is within limits.

Event Code	Message Text and Description	Details and Solution
2004	The Library Startup process failed.	- Verify that magazines are closed, cartridges are fully seated, and that no accessor obstructions exist. - Verify that all modules are powered and any expansion modules are cabled correctly with the interconnect cable. - Verify that a top and bottom cover is properly installed on the library. - Verify that the module alignment locks (at rear of module) are in the proper position. - If the accessor moves front to back, but not vertically, the accessor shipping lock might be positioned incorrectly and must be moved to either the fully locked or fully unlocked position. - If the error persists, review library events for information or restart the library.
2005	Cable to accessor is broken.	Ensure that the spooling cable is fully seated in the base module and connected correctly to the accessor assembly.
2009	Library test failed due to accessor problem.	- Review test requirements and retry the test. - If the test continues to fail, check for accessor obstructions or other accessor problems. - For proper operation, the accessor must be able to reach the bottom of the library. Verify that no obstructions are at the bottom of the library or on the bottom cover of the library in the path of the accessor. - To check for obstructions at the bottom of the library, first power off the library by pressing Power for 5 seconds and select the Default Park location. - When the library is powered off, remove the left magazine of the lowest library module, and verify that the entire area of the bottom cover is free of any objects that might obstruct the accessor's path. - After any obstructions are cleared, replace the magazine, power the library on, and after the library finishes initialization and inventory, verify that no further critical events were generated.
2010	Library test failed due to spooling mechanism defect.	Ensure that the spooling mechanism is fully seated in the base module and installed correctly to the accessor assembly.
2012	Multiple bottom covers detected.	Remove all bottom covers except for the bottom module in the library.
2013	Multiple top covers detected.	Remove all top covers except for the top module in the library.
2014	Bottom cover is missing.	Install the bottom cover on the bottom module of the library, also check the module interconnect cabling and module power cabling. If the base module can't detect both a top and bottom cover, the accessor doesn't move.
2015	Top cover is missing.	Install the top cover on the top module of the library. Also, check the module interconnect cabling and module power cabling. If the base module can't detect both a top and bottom cover, the accessor doesn't move.
2016	Unit to unit not locked.	Ensure that the alignment mechanism is engaged in every module that is above another module in the library.
2017	Communication errors during stack discover process.	- Ensure that all modules are powered and have the interconnect cable properly attached. - Ensure that the module alignment locks (at the rear of module) are in the correct positions.
2021	Database access error.	Restore a configuration backup and run a power cycle.
2022	Drive was hot-removed.	Reinsert the removed drive at the same position as it was removed.
2023	Internal Software error.	Check for a new system Software version for upgrade.
2024	Unhandled Exception.	Check for a new system Software version for upgrade.
2027	Move failed pulling cartridge from slot.	- Check for labels or cartridge misalignments that can prevent the cartridge from coming out of the slot or drive. - For proper operation, the accessor must be able to reach the bottom of the library. Verify that no obstructions exist at the bottom of the library or on the bottom cover of the library in the path of the accessor. - To check for obstructions at the bottom of the library, first power off the library by pressing Power for 5 seconds and select the Default Park location. - When the library is powered off, remove the left magazine of the lowest library module, and verify that the entire area of the bottom cover is free of any objects that might obstruct the accessor's path. - After obstructions are cleared, replace the magazine, power the library on, and after the library finishes initialization and inventory, verify that no further critical events were generated.
2028	Move failed inserting cartridge to slot.	- Check for labels or cartridge misalignments that can prevent the cartridge from coming out of the slot or drive. - For proper operation, the accessor must be able to reach the bottom of the library. Verify that no obstructions exist at the bottom of the library or on the bottom cover of the library in the path of the accessor. - To check for obstructions at the bottom of the library, first power off the library by pressing Power for 5 seconds and select the Default Park location. - When the library is powered off, remove the left magazine of the lowest library module, and verify that the entire area of the bottom cover is free of any objects that might obstruct the accessor's path. - After obstructions are cleared, replace the magazine, power the library on, and after the library finishes initialization and inventory, verify that no further critical events were generated.

Event Code	Message Text and Description	Details and Solution
2029	Initialization failure due to accessor front to back positioning error.	1. Check for obstructions in the pathway of the accessor such as a cartridge that is sticking out. 2. Verify the module alignment and frame alignment. Check whether the accessor is stuck in lock mechanism. 3. Move the accessor apart from lock mechanism and enable lock mechanism correctly.
2032	Initialization failure due to accessor rotation positioning error.	Check for obstructions in the vertical pathway of the accessor, such as a cartridge that is sitting in the shuttle of the accessor or any other impedance to accessor movement.
2033	Initialization failure due to accessor vertical positioning error.	• Check for obstructions in the vertical pathway of the accessor such as a cartridge that is sticking out. • For proper operation, the accessor must be able to reach the bottom of the library. Verify that no obstructions exist at the bottom of the library or on the bottom cover of the library in the path of the accessor. 1. To check for obstructions at the bottom of the library, first power off the library by pressing Power for 5 seconds and select the Default Park location. 2. When the library is powered off, remove the left magazine of the lowest library module, and verify that the entire area of the bottom cover is free of any objects that might obstruct the accessor's path. 3. After obstructions are cleared, replace the magazine, power the library on, and after the library finishes initialization and inventory, verify that no further critical events were generated.
2034	Cable to spooling mechanism is broken during initialization.	Ensure that the spooling mechanism is fully seated in the base module and connected correctly to the accessor.
2035	Initialization failure due to accessor gripper positioning error.	Check for obstructions in the vertical pathway of the accessor, such as a cartridge that is sitting in the shuttle of the accessor or any other impedance to accessor movement.
2036	Unintended process termination.	Restart or power cycle system.
2037	Accessor firmware version upgrade failed.	Restart or power cycle system.
2038	Lost connection to Module.	• Ensure that all modules are powered and have the interconnect cable properly attached. • Restart or power cycle the system.
2039	Cartridge was left in accessor gripper, unable to be moved to any open location.	• Enable I/O station and ensure that empty slots are available in the I/O station. • Power-cycle the library. • If still failing, open covers and remove the cartridge manually from gripper.
2040	Library Verify test failed with critical error.	An unidentified failure occurred. Contact your IBM service representative.
2041	Library Verify test failed because of unit lock failed.	Ensure that the alignment mechanism is engaged in every module that is above another module in the library.
2042	Library Verify test failed because top cover is missing.	• Install the top cover on the top module of the library. • Check the module interconnect cabling and module power cabling. • If the base module can't detect both a top and bottom cover, the accessor doesn't move.
2043	Library Verify test failed because bottom cover is missing.	• Install the bottom cover on the bottom module of the library. • Check the module interconnect cabling and module power cabling. • If the base module can't detect both a top and bottom cover, the accessor doesn't move.
2045	Library Verify test failed because move media test failed.	• Check for obstructions in the pathway of the accessor such as a cartridge that is sticking out. • Verify module alignment and frame alignment. • Check if accessor is stuck in lock mechanism, move the accessor apart from lock mechanism and enable lock mechanism correctly.
2046	Library Verify test failed because drive communication test failed.	• Remove and reseat the drive canister to ensure that the drive is fully seated. • If the issue persists, reset the drive. • Use the library Management GUI to pull a drive support ticket and check the device analysis section for help (HPE Library and Tape Tools must be installed to view support ticket).
2047	Library Verify test failed because the bar code scanning test failed.	• Verify that no obstruction is in front of the bar code scanning module on the cartridge table on the accessor. • If the error persists replace the accessor. • For proper operation, the accessor must be able to reach the bottom of the library. Verify that no obstructions are at the bottom of the library or on the bottom cover of the library in the path of the accessor. 1. To check for obstructions at the bottom of the library, first power off the library by pressing Power for 5 seconds and select the Default Park location. 2. When the library is powered off, remove the left magazine of the lowest library module, and verify that the entire area of the bottom cover is free of any objects that might obstruct the accessor's path. 3. After obstructions are cleared, replace the magazine, power the library on, and after the library finishes initialization and inventory, verify that no further critical events were generated.
2052	An open magazine was detected in one or more modules.	• Ensure that all magazines are inserted and properly locked. • Don't open magazines by using the emergency release while the library is operating and the accessor is moving.

Event Code	Message Text and Description	Details and Solution
2053	An open top cover was detected.	- Ensure that the top cover is inserted and properly locked. - Don't open top cover by using the emergency release while the library is operating and the accessor is moving.
2054	An open bottom cover was detected.	- Ensure that the bottom cover is inserted and properly locked. - Don't open bottom cover by using the emergency release while the library is operating and the accessor is moving.
2055	An open unit lock was detected.	- Ensure that all unit locks are properly locked. - Don't open unit locks by using the emergency release while the library is operating and the accessor is moving.
2056	Initialization failure due to picker push pull positioning error.	Check for obstructions in the horizontal pathway of the accessor such as a cartridge that is sticking out or a cable that is impeding progress.
2057	Startup failure due to shipping lock in incorrect position.	- Get access to the picker assembly and manually move the shipping lock lever to either locked or unlocked position. - After the shipping lock is moved to the one of the correct positions, restart the library.
2061	Move failed pulling cartridge from drive.	Check for labels or cartridge misalignments that would prevent the cartridge from coming out of the drive.
2062	Move failed inserting cartridge to drive.	Check for labels or cartridge misalignments that would prevent the cartridge from moving into the drive.
2063	Move failed positioning picker in front of drive.	Check for obstructions in the vertical or horizontal pathway of the accessor. Examples might include a cartridge that isn't seated completely in a slot, an accessor isn't sitting horizontally level, or a problem with the accessor spooling cable that is impeding progress.
2064	Library test failed with critical error.	An unidentified failure occurred. Contact your IBM service representative.
2066	Library startup process failed during inventory scan.	- Verify that magazines are closed, cartridges are fully seated, and that no accessor obstructions exist. - Verify that all modules are powered and any expansion modules are cabled correctly with the interconnect cable. - Verify that a top and bottom cover is properly installed on the library. - Verify that the module alignment locks (at rear of module) are in the proper position. - If the accessor moves front to back, but not vertically, the accessor shipping lock might be positioned incorrectly and must be moved to either the fully locked or fully unlocked position. - If the error persists, review library events for information or restart the library.
2067	For safety reason, the accessor movement was halted in place.	- Ensure that all magazines, top or bottom covers and unit locks are inserted and properly locked. - Don't open magazines by using the emergency release or remove covers or unit locks while the library is operating and the accessor is moving. - Ensure that all modules are powered and have the interconnect cable properly attached.
2068	An emergency stop condition was detected in one or more modules and prevented the accessor from initialization.	- Ensure that all magazines, top or bottom covers and unit locks are inserted and properly locked. - Insert all open magazines and install all necessary covers and unit locks before the library is powered on. - Ensure that all modules are powered and have the interconnect cable properly attached.
2069	Initialization failure due to bar code reader error.	Restart the library and if the error persists, replace the accessor assembly.
2070	Inventory scan failed because of Elevator axis problem.	- Check for obstructions in the vertical pathway of the accessor such as a cartridge that is sticking out. - Verify module alignment and frame alignment. - For proper operation, the accessor must be able to reach the bottom of the library. Verify that no obstructions are at the bottom of the library or on the bottom cover of the library in the path of the accessor. - To check for obstructions at the bottom of the library, first power off the library by pressing Power for 5 seconds and select the Default Park location. - When the library is powered off, remove the left magazine of the lowest library module, and verify that the entire area of the bottom cover is free of any objects that might obstruct the accessor's path. - After obstructions are cleared, replace the magazine, power the library on, and after the library finishes initialization and inventory, verify that no further critical events were generated.
2071	Cartridge on picker when trying to scan.	- Verify that no obstruction is in front of the bar code scanning module on the cartridge table that is on the accessor. - If the error persists, replace the accessor.
2072	Bottom cover was detected at an incorrect position.	Review the stack assembly and place the covers to the proper position.
2073	Top cover was detected at an incorrect position.	Review the stack assembly and place the covers to the proper position.
2074	The library startup failed due to a GPIO error.	Restart or power cycle system.
2075	The library startup failed due to an error when trying to open the accessor serial port.	Restart or power cycle system.

Event Code	Message Text and Description	Details and Solution
2076	I2C bus signals invalid.	- Remove all drive canisters of the affected chassis and restart the library. - If the problem persists, replace the chassis. If not, add one drive after the other until the problem comes back. - Replace the last drive that was added before it failed again.
2077	Failed to store Calibration Data to Chassis.	Restart or power cycle system. Note: This is a singular ticket and must be resolved manually.
2078	Incompatible Robotics Assembly without Encoder detected	Replace Robotics Assembly with a compatible model with Encoder or upgrade Firmware to a version that supports Encoder-less control.
2079	Couldn't upgrade bar code reader firmware.	Restart the library and if the error persists, replace the accessor.
2080	Cartridge lost while inserting it into slot/drive.	Check the source/destination element and ensure that no obstructions are in the pathway of the accessor.
2085	Communication failure to the Base Module controller board I2C port expander component	This event is reported if read or write access to I2C port expanders on the main library controller is failing. - Reboot the library to see if the error persists. - If the error persists, power off the library and reseal the base module controller. - If the error continues to persist, replace the base module controller.
2086	Communication failure to the Expansion Module controller board I2C port expander component.	This event is reported if read or write access to I2C port expanders on expansion controller is failing. - Restart the library to see if the error persists. - If the error persists, power off the library and reseal the expansion module controller. - If the error continues to persist, replace the expansion module controller.
2087	Error accessing the backplane flash memory.	- Restart the library and if the error persists, replace the chassis. - Before the chassis is replaced, ensure that you remove all of your tape cartridges. - If magazines need to be removed to get access to the tape cartridges, first power down the device and then manually release each magazine. Only one magazine must be opened at a time.
2089	Incompatible Robotics Assembly Detected	An incompatible robotics assembly is detected. The robotics assembly was not powered on to avoid damage to the library. Power off the library and replace the robotics assembly with a compatible version.
2092	Locking the Robotics Assembly has failed during power off process	Power up the library. In case of robotics failure, error events are reported at power-up. Look for the proposed solution for these new errors. If no new errors are reported, the 2092 event can be ignored.
2093	Communication to accessor controller couldn't be established.	Restart the library and if the error persists, replace the accessor assembly.
2094	An emergency stop condition was detected in one or more modules and prevented the accessor from running the inventory scan.	- Ensure that all magazines, top or bottom covers and unit locks are inserted and properly locked. - Insert all open magazines and install all necessary covers and unit locks before powering on the library. - Ensure that all modules are powered and have the interconnect cable properly attached.
2095	Inventory scan failed because of accessor positioning problem.	Check for obstructions in the horizontal pathway of the accessor such as a cartridge that is sticking out or lying on the accessor table.
2096	Initializing a communication interface on the library controller failed.	Restart the library, and if the error persists, replace the library controller.
2097	Accessor reinitialization failed.	Restart the library and if the error persists, replace the accessor assembly.
2100	Robotic move to requested position failed.	- Check for obstructions in the pathway of the accessor such as a cartridge that is sticking out. - Verify module alignment and frame alignment. - Check if the accessor is stuck in the lock mechanism. Move the accessor away from the lock mechanism and enable the lock mechanism correctly.
2103	Incorrect stack assembly, too many expansion modules below main library.	Ensure that no more than three expansion modules are mounted and connected below or above main library.
2104	Incorrect stack assembly, too many expansion modules above main library.	Ensure that no more than three expansion modules are mounted and connected below or above main library.
2105	Accessor initialization failed due to horizontal positioning problem.	- Check for obstructions in the horizontal pathway of the accessor such as a cartridge sticking out or lying on the accessor's table. - The accessor shipping lock might be positioned incorrectly and must be moved to either the fully locked or fully unlocked position.
2106	An elevator block was detected and as a result the system was taken offline.	- Check for obstructions in the vertical pathway of the robot, such as a cartridge sticking out. Also, verify alignment of the module and the frame. - Follow these steps to check for any obstructions at the bottom of the library or on the bottom cover of the library: - Turn off the library by pressing the front power button for 5 seconds, and select Default Park location. - Remove the left magazine of the lowest library module. - Remove any objects from the bottom cover that might obstruct the robot's path. - Replace the magazine. - Power on the library. - After the library finishes initialization and inventory, verify that no further critical events were generated.

Warning error events

An appropriate message is posted on the Operator Control Panel and the Management GUI.

Table 1. Warning events

Event Code	Message Text and Description	Details and Solution
4000	Drive Sled Fan Speed too low.	Ensure that no obstructions are in the fan.
4002	Drive clean request.	Clean the drive with an approved cleaning cartridge.
4003	The drive configuration failed.	Remove and reseal the drive canister and retry the operation.
4004	The drive status request failed.	Remove and reseal the drive canister to ensure that the drive is fully seated. If the issue persists, reset the drive.
4005	Drive is reporting a critical TapeAlert.	Power-cycle the drive, and verify whether the drive reports the same TapeAlert.
4006	A drive reported temperature is above the threshold.	- Verify that the drive fan is spinning, isn't obstructed, and that the ambient temperature is within specification. - Ensure that drive bay cover plates are in place in each location where no drive is installed. The drive cover plates are required for proper airflow.
4008	Cleaning tape expired.	Discard the cleaning cartridge and retry the cleaning operation with a new cleaning cartridge.
4009	Firmware upgrade of one or multiple expansion modules failed.	The Base Module must be able to communicate with a powered on and connected expansion module to complete the upgrade. - Reseat the expansion controller and check the interconnect cable and power connections. - Retry the firmware upgrade.
4010	Incompatible drive.	Remove the incompatible drive. Install only drives that are supported by the library.
4012	Move Cartridge failed due to cartridge issue.	- View the event details to determine which cartridge was involved. - Verify surrounding events that might point to problems with this media in other move operations. - Remove the media from the library, and physically inspect the media to ensure that no physical damage exists. - If the media appears to be undamaged, put the media back into the library and retry the move operation. If the problem persists, retry the operation with a different cartridge in the same drive. - If the problem follows the media, remove the media from use. - If the problem follows the drive, use the library Management GUI to pull a drive support ticket and check the device analysis section for help.
4014	Library test failed due to a Drive issue.	- Verify the test parameters, and retry the test. - If the test fails, check the library event log for specific events that are associated with this drive. - Use the Management GUI to pull a drive support ticket and check the device analysis section for help.
4015	The Power system is degraded. Redundancy is not available.	Ensure that all power supplies are installed properly (up to two per module), and that each power supply is connected to a valid power source.
4016	Back up configuration data to base module failed.	Attempt to save the library configuration, power cycle the library, and retry the operation.
4017	Restore configuration data from Chassis failed.	Attempt to save the library configuration, power cycle the library, and retry the operation.
4019	Drive Firmware bundle upgrade failed.	- Verify that the firmware file is correct for the drives. - Ensure that the drives are in a healthy state with no cartridge in the drive. - Retry the operation.
4020	Database was reset due to a problem that prevented the library from powering up.	- If the library was restored to default settings, restore a saved configuration by using a previously saved config file. - If no config file exists, then proceed in configuring the library.
4021	Drive was hot-removed while in active status as data transfer device.	Tape drives must be disabled (powered-off) before they are removed from the Library. In case of hot removal, this event is created if the drive was no active LUN master. A similar event is defined with higher severity (critical) if the drive was LUN master. - Put the drive back into the library. - Follow the Removing a tape drive procedure to remove the tape drive.
4022	Drive in wrong position.	A full height drive was installed to a drive bay with even physical number. These positions can be used only with half height drives because elements of the parting wall are covering the medium load area.
4025	Library test failed due to a cartridge error.	Remove the cartridge and inspect it for damage. Retry operation with another cartridge.
4028	Library Test failed due to incompatible cartridge.	Check LTO generation for cartridge and drives. Remove cartridges that aren't compatible to your tape drives.
4029	Library Test failed - Cartridge bar code indicates incompatibility with drive.	Check if Media bar code label is matching LTO generation. Replace the label or remove incompatible media from your system.
4030	Move cartridge operation failed due to cartridge error.	Remove the cartridge and inspect it for damage. Retry operation with another cartridge.
4041	Library Verify failed because the power supply redundancy test failed.	Ensure that all power supplies are installed properly (two per module), and that each power supply is connected to a valid power source.

Event Code	Message Text and Description	Details and Solution
4044	One of the Library tests failed because a source element or destination element isn't accessible.	Verify the source and destination elements and retry the move operation.
4060	Connection to the KMIP server failed.	- Verify the username and password and all needed SSL certificates that are needed for connecting to the KMIP server. - Verify that the KMIP server is reachable within the network. - Verify the IP addresses and host names of the KMIP servers that are entered into the wizard. If this error happens during the connectivity check, the additional information parameter includes further error information that is received from the Cryptsoft KMIP library. It also includes the information if the server wasn't able to ping.
4061	Key is not found on KMIP server.	Verify that the requested key is available on the KMIP server. Check the KMIP server logs for details.
4062	Key creation on KMIP server failed.	Check the KMIP server logs for details about why key creation failed.
4063	KMIP configuration invalid.	Use the KMIP configuration wizard to verify the KMIP configuration.
4064	KMIP feature not licensed.	Disable KMIP or install appropriate license for KMIP feature.
4065	A tape alert flag was reported by a drive.	Look for logged TapeAlert flags and see its description in the TapeAlert flags section.
4067	Cleaning cartridge will soon expire and must be replaced.	Replace the cartridge.
4072	No cleaning cartridge in logical library available for auto cleaning.	- Auto cleaning is enabled, but the logical library contains no labeled cleaning cartridge. The library was unable to complete the auto clean function for one or more drives in this logical library. - Install a valid and labeled cleaning cartridge into the logical library and then complete a load and unload on the drive that needs cleaning to start the auto cleaning.
4073	Medium source element empty.	Check the source slot visually and rescan the inventory. Additionally, check for valid and readable bar code label.
4074	Medium source element empty.	Check the source slot visually and rescan the inventory. Additionally, check for valid and readable bar code label.
4075	Cartridge lost while it was extracted from slot/drive.	Check the source/destination element and ensure that no obstructions are in the pathway of the accessor.
4077	Unlocking the right magazine failed.	- Check if any obstacle is preventing the robot from movement. - Restart the library and retry the operation. - If the error persists, replace the chassis. - If the magazine needs to be removed to get access to the tape cartridges, first power down the device, and then release the magazine manually. Only one magazine can be open at a time.
4078	Unlocking the left magazine failed.	- Check if any obstacle is preventing the robot from movement. - Restart the library and retry the operation. - If the error persists, replace the chassis. - If the magazine needs to be removed to get access to the tape cartridges, first power down the device, and then release the magazine manually. Only one magazine can be open at a time.
4079	Unlocking the I/O station failed.	- Check if any obstacle is preventing the robot from moving. - Reset the library and retry the operation. - If the error persists, replace the chassis. - If the I/O station needs to be removed to get access to the tape cartridges, first power down the device, and then release the magazine manually. Only one magazine can be opened at a time.
4080	Library Verify test failed with warning.	An unidentified failure occurred. Contact your IBM service representative.
4085	Drive command retries expired.	- Run a drive read/write test (Library > Actions > Tests > Drive test) with a known working cartridge. - If the drive test is successful, check in the host for errors that are related with the retries. - If the drive test failed, replace the drive.
4086	Move operation failed.	- Ensure that the network the library is connected to is operating normally. - Ensure that the library is running the current firmware. - Restart the library.
4089	Auto calibration of one or more modules failed. Adjustment to calibration target failed.	- The library must be re-calibrated. - Ensure that the library firmware is up-to-date. - This event indicates that one or more of the gray calibration targets on the library magazines couldn't be used in calibration. - Inspect the calibration targets in each module and then repeat the auto-calibration routine with the Management GUI.
4090	Auto calibration of one or more modules failed. Calibration target not found.	- The library must be re-calibrated. - Ensure that the library firmware is up-to-date. - This event indicates that one or more of the gray calibration targets on the library magazines could not be used in calibration. - Inspect the calibration targets in each module and then repeat the auto-calibration routine with the Management GUI.
4091	Auto calibration of one or more modules failed. Adjustment out of range.	- The library must be re-calibrated. - Ensure that the library firmware is up-to-date. - This event indicates that one or more of the gray calibration targets on the library magazines could not be used in calibration. - Inspect the calibration targets in each module and then repeat the auto-calibration routine with the Management GUI.

Event Code	Message Text and Description	Details and Solution
4093	Could not obtain an IP address from DHCP server.	- Check the network configuration settings and check if the DHCP server is reachable. - Use the network configuration menu or unplug the network cable and plug it in after a few seconds to trigger an automatic reconfiguration of the network interface.
4095	Library test failed. Not enough valid cartridges available for testing.	Load the cartridges into the library.
4098	System time synchronization with SNTP failed.	Check for valid SNTP server address in Time configuration. If correct, ensure that the server is reachable from your network and not blocked by a firewall.
4099	An unexpected reset of accessor was detected.	Ensure that the spooling cable is fully seated in the base module and connected correctly to the accessor assembly. If the error recurs, replace the accessor assembly.
4112	Move Cartridge failed due to cartridge not seating properly.	- Verify surrounding events for drive problems. - Retry the operation with the same source and destination combination. - If the problem persists, retry the operation with a different cartridge in the same drive. If the problem follows the cartridge, the cartridge must be checked for physical damage and no longer used. - Use the library Web GUI to pull a drive support ticket and check the device analysis section for more help.
4113	Move from drive failed.	Check for labels or cartridge misalignments that prevents the cartridge from coming out of the slot or drive.
4117	Drive disabled because no power supply available.	Remove all affected drives, insert, and power up at least one power supply to the failing module. Wait 10 seconds and put the drives back into the module.
4119	Drive disabled because internal IP address unknown.	- Remove affected drive, wait 10 seconds, and put it back into the module. - Restart Library stack.
4120	No empty drive available for system test.	Make sure that at least one empty drive and one compatible cartridge are available.
4121	No compatible media available for system test.	Make sure that at least one empty drive and one compatible cartridge are in the drive.
4122	No cartridge available for slot to slot exerciser test.	Make sure that at least one cartridge and one empty slot are in the library.
4123	No empty slot available for slot to slot exerciser test.	Make sure that at least one cartridge and one empty slot are in the library.
4124	Drive or media statistics couldn't be retrieved when the tape was unloaded.	Check for more warning tickets. Replace media if media-related tape alert flags reported.
4126	Cartridge was found in inaccessible slot of lowermost unit.	Bottom slots from the lowermost unit are inaccessible to the accessor. Place the cartridge in an upper slot.
4127	Drive was restarted because of canister reset.	Verify that the drive is installed properly in its slot and thumb screws are tightened. If the error persists, replace the drive.
4128	An installed power supply is detected but doesn't provide power.	Ensure that the power supply has a power cord plugged in and is connected to a valid power source. Although the power source is not available, this expansion module can still be used for tape storage. Operation of tape drives is not possible.
4129	Move from drive failed.	Check backup application how to allow media removal from drive. If unsuccessful, try the Force Drive Media Eject option in the Operations menu.
4133	Protection Foam not removed from Base Module	- Power down the library. - Remove top cover and then remove the protection foam. - Install the top cover again and restart the library.
4135	Drive diagnostic failed.	Use another diagnostic tape and run test once more. If still failing, download drive dumps and contact service.
4136	The base module detected an installed power supply but this power supply doesn't provide power.	Ensure that the power supply has a power cord plugged in and is connected to a valid power source.
4137	Diagnostic Tape not removed.	Open Magazine or I/O station to remove the diagnostic tape with the OP or Management GUI.
4138	USB over-current detected. The USB port was disabled.	Ensure that the USB device is correctly inserted and that it does not consume more current than specified in manual.
4139	Magazine or I/O Station operation failed.	- Check whether any obstacle is preventing the accessor from movement. - Restart the library and retry the operation. If the error persists, replace the accessor assembly. - If the magazine needs to be removed to get access to the tape cartridges, first power down the device and then release the magazine manually. Only one magazine can be open at a time.
4140	Personality mismatch detected.	Replace either chassis or library controller to ensure that all parts in the stack are matching the personality of the main library controller.
4142	Medium destination element full.	Ensure that your destination slot or drive is empty and try again.
4144	Unit to unit lock of lowermost module is engaged.	Ensure that the alignment mechanism isn't engaged in the lowermost module.
4148	Download of one or multiple drive dumps failed	Check status of selected drives and ensure that they are present and finally initialized before downloading dumps.
4150	Sequential Mode move operation failed.	Check event details for further information.
4151	Download of drive firmware image completed, but firmware revision did not change after restart.	- Check if the uploaded firmware image is matching your drive type and generation. - Ensure that the image file isn't corrupted. - Download a new image from the drive vendor's website if you aren't sure about file integrity.
4152	KMIP Connection refused.	The selected port on the target machine is not open. The connection is refused. Check that the server application is running on the target machine and the firewall is not blocking the selected port. Contact IT Personnel to verify the port settings.

Event Code	Message Text and Description	Details and Solution
4153	KMIP Server does not trust the client certificate.	Use a client certificate that is signed by a trusted certificate authority (CA) or manually select the untrusted certificate on the server side and trust it (not available on all servers).
4154	KMIP Ping to server failed.	The target machine could not be reached, so no network connection possible. - Verify that the IP address in the settings is correct. - Check that the target machine is powered and connected to the network. - Check the network cable. - Verify that the Firewall setting on the target machine allows ping requests and responses.
4155	KMIP No route to host.	The target machine could not be reached. The network route to the machine isn't available. - Check your IP settings (IP address, Gateway and Netmask) and verify it with your IT personnel. - Check that the Firewall settings on the target machine are correct.
4156	KMIP Handshake failure.	The TLS connection could not be established because of Handshake errors during certificate exchange. - Check the certificates on server and client side for valid entries and that they are still valid and not expired. - Verify that TLS1.2 is enabled on the server. - Check the client and server date/time for current time. - Ask your IT personnel for new and valid certificates.
4157	KMIP Certificate unknown.	The server certificate is unknown because its root certificate is missing or not trusted. Run a new certificate request with your server or certificate authority and import the resulting certificate chain.
4158	KMIP host name lookup failed.	The host name on the network could not be found. It does not exist or is misspelled. - Verify that the entered host name is correct. - Check your DNS address in the network settings. - Contact your IT personnel to verify the entered data.
4159	KMIP certificate verify failed.	The TLS server certificate could not be verified as a valid and trusted certificate. - Check if your server root certificate changed. - Create a certificate request against your server to generate a new client certificate based on the changed server certificates.
4160	Connection to SKLM/GKLM server failed	Retrieving of IPP encryption key failed because connection to SKLM/GKLM server could not be established. Check for availability of server and verify your server settings. Add a secondary fallback server if possible.
4161	Internal encryption key handshake failed	Retrieving IPP encryption key failed because of internal data transfer failure between drive and library. Verify if the latest drive software is installed. If the failure reoccurs, check the library event log for specific events associated with this drive.
4162	for z/OS encryption not licensed	Disable SKLM/GKLM for z/OS encryption or install appropriate license for Library Managed Encryption.
4163	Drive sled discovery timeout	Drive sled discovery timeout, status of Drive sleds not available in time. - Ensure that all modules are powered and have the interconnect cable properly attached. - If this event is seen on multiple modules or after ensuring all interconnect cables are properly attached, ensure that the network that the base module is connected to is not experiencing broadcast storms or other abnormal activity. - Reboot or power cycle the system to rediscover the modules.
4164	Inventory has been updated due to an unexpected empty or full slot.	If a move fails due to an unexpected empty or full slot, the slot is rescanned and the inventory is corrected.
4165	Bottom magazine slots in the lowermost unit are not accessible.	The installed accessor does not support access to all 40 slots in the lowermost unit. The bottom slots in the lowermost unit are not accessible, so only 32 slots are available. Install an accessor that supports access to all 40 slots in the lowermost unit.
4166	Drive self-test diagnostic failed	Reset Drive and run the test again. If still failing, download drive dumps and contact service.
4167	Drive primary port diagnostic failed	- Check whether the wrap test connector is correctly connected to the selected port. - Reset drive and run test again. - If the diagnostic still fails, download drive dumps and contact service.
4168	Drive encryption key path diagnostic failed.	- Check the SKLM server configuration and your network settings. - If the diagnostic still fails, download the library logs and contact service.
4173	Encryption Key Path diagnostics failed.	- Check the Key server configuration and your network settings. - If the diagnostic still fails, download the library logs and contact service.
4174	KMIP CA certificate failure.	Check whether you provided the correct CA certificate or the CA certificate on the encryption server was changed.
4175	Failed to create default logical library.	Check that no errors occurred during the startup and that at least one drive is present in the library.

Configuration Change events

Table 1. Configuration Change events

Event Code	Message Text and Description
8000	The configuration of a drive changed.

Event Code	Message Text and Description
8001	The drive was added or removed from the system.
8002	A logical library was added/removed or changed.
8003	I/O station was enabled/disabled.
8004	Drive firmware changed due to firmware upgrade.
8005	Host name/domain name changed.
8006	Email configuration settings changed.
8007	Date/time format changed.
8009	Time zone configuration changed.
8011	Network configuration changed.
8012	Expansion Module upgraded.
8013	NTP time server settings changed.
8014	The SSH Access was enabled/disabled.
8016	Library reset default settings started by user.
8017	Library Firmware changed.
8018	The Unlabeled Cartridge Support configuration changed.
8019	Accessor firmware version upgraded.
8022	Management GUI/Operator Panel Timeout configuration changed.
8024	I/O station/Magazine access control configuration changed.
8026	Accessor change detected.
8029	The SNMP configuration changed.
8030	An SNMP target was added.
8031	An SNMP target was deleted.
8033	The Operator Panel module was changed.
8034	Manual Drive reset executed.
8036	New chassis detected.
8037	Chassis was removed.
8040	LDAP Server was added.
8041	LDAP Server was modified.
8042	LDAP Server was deleted.
8043	LDAP User was added.
8044	LDAP User was modified.
8045	LDAP User was deleted.
8046	Logout prevention configuration changed.
8057	Hardware component added.
8058	Hardware component removed.
8059	Hardware component of Library replaced.
8060	New Expansion Controller detected.
8061	New Base Library Controller detected.
8062	Auto calibration successfully finished.
8064	Password rules configuration changed.
8065	User was added.
8066	User was deleted.
8068	Remote Logging configuration changed Note: Not applicable for 6U WHITEBOX
8069	User password changed
8071	Administrator permission for configuring encryption changed
8072	Kerberos Realm was added
8073	Kerberos Realm was modified
8074	Kerberos Realm was deleted
8075	Kerberos User was added
8076	Kerberos User was modified
8077	Kerberos User was deleted
8079	SKLM/GKLM for z/OS Encryption settings modified
8085	LDAP User Group was added
8086	LDAP User Group was modified
8087	LDAP User Group was deleted
8088	Logical Library/Partition modification and creation settings changed

Informational events

Table 1. Informational Events

Event Code	Message Text and Description
9000	A tape alert flag was reported by a drive.
9001	A drive is present but currently disabled.
9002	The library was powered on.

Event Code	Message Text and Description
9003	Move Cartridge command was run.
9004	Inventory scan was completed.
9005	The library was powered down from Front Panel.
9006	The network interface was turned on.
9007	The network interface was turned off.
9008	The System Time was synchronized with an NTP server.
9009	A magazine was unlocked and opened.
9010	A magazine was closed and locked.
9011	An I/O station was unlocked and opened.
9012	An I/O station was closed and locked.
9013	A user logged in at the Management GUI.
9014	A user logged out at the Management GUI.
9015	A user logged in at the Operator Panel interface.
9016	A user logged out at the Operator Panel interface.
9024	Drive support ticket created.
9025	Library test started.
9026	Library test successfully finished.
9027	Library test was stopped by user.
9028	Configuration back up to chassis was successful.
9029	Configuration restore from chassis was successful.
9031	Library health Status was changed to status "OK".
9032	Library health status was changed to status "Warning".
9033	Library health status was changed to status "Critical".
9035	New library chassis detected.
9038	The library was rebooted through user interface.
9041	Key on KMIP server created.
9043	Drive cleaning was started.
9045	Library configuration data failed to duplicate on to the Base Module.
9060	One or multiple configured DNS servers aren't responding.
9061	User account was locked due to too many invalid login attempts on Management GUI.
9062	Invalid password used for login.
9063	The network port used to contact the encryption server was changed.
9064	Backup of certificate created.
9065	Certificate is restored.
9066	Temporary password generated.
9067	LTO7 formatted cartridge with a Type M bar code detected.
9068	Type M cartridge without a Type M bar code detected.
9069	Sequential Mode load sequence ended because last storage cartridge of logical library was unloaded.
9070	Sequential Mode load sequence restarted (Loop Mode) because last storage cartridge of logical library was unloaded.
9077	User submitted feedback
9078	Unexpected drive reset occurred
9079	Media optimization needed

TapeAlert flags

This section is intended to provide information to the reader about the tape drive by using TapeAlert technology.

All error code and diagnostic information can be accessed from the Management GUI of the library. The drive portion of the Management GUI contains drive error codes. Therefore, it is not necessary to open the Library to access the buttons on the drive. See [Locating Management functions](#) for a complete description of the Management GUI functions and displays.

TapeAlert is a standard that defines status conditions and problems that are experienced by devices such as tape drives, autoloaders, and libraries. The standard enables a server to read TapeAlert messages (called flags) from a tape drive with the SCSI bus. The server reads the flags from Log Sense Page 0x2E. Refer to the *IBM TS4300 Tape Library SCSI Reference* for library and drive TapeAlert Flag information.

This library is compatible with TapeAlert technology, which provides error and diagnostic information about the drives and the library to the server. Because library and drive firmware might change periodically, the SNMP interface in the library does not require code changes if devices add more TapeAlerts that are not supported today. However, if this issue occurs the Management Information Block (MIB) is written to minimize impact to the SNMP monitoring station. At the time of this writing, the TapeAlert flags correctly represent TapeAlerts that are sent. The MIB file must not be taken to mean that all traps that are defined in the MIB are sent by the library or that they will be sent in the future.

- [TapeAlert flags supported by the library](#)
- [TapeAlert flags supported by the drive](#)

TapeAlert flags supported by the library

Parameter Code	Flag name	Type	Description
01d	Library Hardware A	C	The media changer mechanism is having difficulty communicating with the drive: • Turn the media changer OFF, then ON • Restart the operation. • If problem persists, contact Technical Support.
02d	Library Hardware B	W	There is a problem with the media changer mechanism. If the problem persists, contact Technical Support.
04d	Library Hardware D	C	The library has a hardware fault that is not mechanically related or requires a power cycle to recover. • Turn the media changer OFF, then ON. • Restart the operation. • If the problem persists, contact Technical Support.
05d	Library Diagnostics Required	W	The library mechanism might have a hardware fault. Run extended diagnostics to verify and diagnose the problem. Check the library user's manual for device-specific instructions on running extended diagnostic tests.
13d	Library Pick Retry	W	There is a potential problem with the drive ejecting cartridges or with the library picking cartridges from a slot. • No action needs to be taken at this time. • If the problem persists, contact Technical Support.
14d	Library Place Retry	W	There is a potential problem with the library mechanism placing a cartridge into a slot. • No action needs to be taken at this time. • If the problem persists, contact Technical Support.
15d	Library Load Retry	W	There is a potential problem with the drive or the library mechanism loading cartridges, or an incompatible cartridge. This flag is cleared when the next move command is received.
16d	Library Door	C	The operation failed because the library door is open. • Clear any obstructions from the library door. • Close the library door. • If the problem persists, call the library supplier help line.
17d	Library I/O Station	C	There is a mechanical problem with the library media I/O Station.
19d	Library Security	W	Library security is compromised. The door was opened then closed during operation.
20d	Library Security Mode	I	The library security mode was changed. The library was either put into secure mode, or the library exited the secure mode. This is for information purposes only. No action is required.
21d	Library Offline	I	The library was manually turned offline and is unavailable for use.
22d	Library Drive Offline	I	A drive inside the library was taken offline. This is for information purposes only. No action is required.
24d	Library Inventory	C	The library detected an inconsistency in its inventory. • Redo the library inventory to correct inconsistency. • Restart the operation.
28d	Power Supply	W	A redundant power supply failed inside the library. Check the library users manual for instructions on replacing the failed power supply.
33d	Library Capacity Exceeded	C	The total number of volumes exceeds the available number of storage elements. Remove a cartridge from the inventory to recover.
• I = Informational suggestion to user • W = Warning. Remedial action is advised. Performance of data might be at risk. • C = Critical immediate remedial action is required.			

TapeAlert flags supported by the drive

Flag Number	Flag Name	Hex Code	Description	Action Required	Event
1	Read warning	01h	Set when the tape drive is having problems reading data. No data is lost, but there is a reduction in the performance of the tape.	Isolate the fault between drive and tape by following these steps: • Use a known good tape cartridge in the suspect drive. If the drive fails, contact your IBM® Service Representative. • Use the suspect tape cartridge in a known good drive. If the test fails, discard the cartridge.	Warning Event

Flag Number	Flag Name	Hex Code	Description	Action Required	Event
2	Write warning	02h	Set when the tape drive is having problems writing data. No data is lost, but there is a reduction in the performance of the tape.	Isolate the fault between drive and tape by following these steps: - Use a known good tape cartridge in the suspect drive. If the drive fails, contact your IBM® service representative. - Use the suspect tape cartridge in a known good drive. If the test fails, discard the cartridge.	Warning Event
3	Hard error	03h	Set for any unrecoverable read, write, or positioning error. (This flag is set with flags 4, 5, or 6).	See the Action Required column for Flag Number 4, 5, or 6 in this table. Ensure that tape drive firmware is at the latest version. See Minimum firmware levels for common library features .	Warning Event
4	Media	04h	Set for any unrecoverable read, write, or positioning error that is due to a faulty tape cartridge.	Replace the tape cartridge.	Warning Event
5	Read failure	05h	Set for any unrecoverable read error where isolation is uncertain and failure might be due to a faulty tape cartridge or to faulty drive hardware.	If Flag Number 4 is also set, the cartridge is defective. Replace the tape cartridge. If Flag Number 4 is not set, see Error Code 6 in Drive Error Codes: Single-character display (SCD) .	Warning Event
6	Write failure	06h	Set for any unrecoverable write or positioning error where isolation is uncertain and failure might be due to a faulty tape cartridge or to faulty drive hardware.	If Flag Number 9 is also set, make sure that the write-protect switch is set so that data can be written to the tape. If Flag Number 4 is also set, the cartridge is defective. Replace the tape cartridge. If Flag Number 4 is not set, see Error Code 6 in Drive Error Codes: Single-character display (SCD) .	Warning Event
7	Media life	07h	Set when the tape cartridge reaches its end of life (EOL).	- Copy the data to another tape cartridge. - Discard the old (EOL) tape.	Warning Event
8	Not data grade	08h	Set when the cartridge is not data-grade. Any data that you write to the tape is at risk.	Replace the tape with a data-grade tape.	Warning Event
9	Write protect	09h	Set when the tape drive detects that the tape cartridge is write-protected.	Make sure that the cartridge's write-protect switch is set so that the tape drive can write data to the tape.	Warning Event
10	No removal	0Ah	Set when the tape drive receives an UNLOAD command after the server prevented the tape cartridge from being removed.	Refer to the documentation for your server's operating system.	Info Event
11	Cleaning media	0Bh	Set when you load a cleaning cartridge into the drive.	No action is required. Informational message only.	Info Event
12	Unsupported format	0Ch	Set when you load an unsupported cartridge type into the drive or when the cartridge format is corrupted.	Use a supported tape cartridge.	Info Event
14	Unrecoverable snapped tape	0Eh	Set when the tape is snapped/cut or has a mechanical failure.	Do not attempt to extract the old tape cartridge. Call the tape drive supplier's help line.	Warning Event
15	Cartridge memory chip failure	0Fh	Set when a cartridge memory (CM) failure is detected on the loaded tape cartridge.	Replace the tape cartridge. If this error occurs on multiple cartridges, see Error Code 6 in Drive Error Codes: Single-character display (SCD) .	Warning Event
16	Forced eject	10h	Set when you manually unload the tape cartridge while the drive was reading or writing.	No action is required. Informational message only.	Warning Event
17	Loaded media is Read-only format	11h	Set when a write attempt is made on a read-only cartridge. The flag is cleared when the cartridge is ejected (this flag is not supported for Ultrium 1 or Ultrium 2).	No action is required. Informational message only.	Warning Event
18	Tape directory is corrupted in the cartridge memory	12h	Set when the drive detects that the tape directory in the cartridge memory is corrupted.	Reread all data from the tape to rebuild the tape directory.	Warning Event
19	Nearing media life	13h	Set when the tape cartridge is nearing its specified end of life.	- Copy the data to another tape cartridge. - Replace the tape cartridge.	Info Event
20	Clean now	14h	Set when the tape drive detects that it needs cleaning.	Clean the tape drive.	Warning Event
21	Clean periodic	15h	Set when the drive detects that it needs routine cleaning.	Clean the tape drive as soon as possible. The drive can continue to operate, but you must clean the drive soon.	Warning Event
22	Expired clean	16h	Set when the tape drive detects an expired cleaning cartridge.	Replace the cleaning cartridge.	Warning Event
23	Invalid cleaning tape	17h	Set when the drive expects a cleaning cartridge and the loaded cartridge is not a cleaning cartridge.	Use a valid cleaning cartridge.	Warning Event

Flag Number	Flag Name	Hex Code	Description	Action Required	Event
25	Interface	19h	Set when the tape drive detects a problem with the SCSI, Fibre Channel, or RS-422 interface.	Locate Error Code 8 or 9 in Drive Error Codes: Single-character display (SCD) .	Warning Event
26	Cooling Fan Failure	1Ah	A tape drive cooling fan failed.	Fan failure inside tape drive mechanism or tape drive enclosure.	Warning Event
27	Power Supply	1Bh	A redundant power supply failed inside the tape drive enclosure. Check the enclosure users manual for instructions on replacing the failed power supply.	A redundant power supply failed inside the tape drive enclosure. Check the enclosure users manual for instructions on replacing the failed power supply.	Warning Event
30	Hardware A	1Eh	Set when a hardware failure occurs that requires that you reset the tape drive to recover.	A redundant power supply failed inside the tape drive enclosure. Check the enclosure users manual for instructions on replacing the failed power supply.	Warning Event
31	Hardware B	1Fh	Set when the tape drive fails its internal Power-On Self-Tests.	Note the error code on the single-character display and see in Drive Error Codes: Single-character display (SCD) for the appropriate instructions.	Warning Event
32	Interface	20h	Set when the tape drive detects a problem with the SCSI, Fibre Channel, or RS-422 interface.	Set when the tape drive detects a problem with the SCSI, Fibre Channel, or RS-422 interface.	Warning Event
33	Eject media	21h	Set when a failure occurs that requires you to unload the cartridge from the drive.	Unload the tape cartridge, then reinsert it and restart the operation.	Warning Event
34	Download fail	22h	Set when the tape drive detects a problem with the SCSI, Fibre Channel, or RS-422 interface.	Ensure that it is the correct FMR image. Download the FMR image again.	Warning Event
35	Drive humidity	23h	Sets when the drive humidity sensor indicates that the drive's humidity exceeds the recommended humidity of the drive.	See Error Code 1 in Drive Error Codes: Single-character display (SCD) .	Warning Event
36	Drive temperature	24h	Set when the drive's temperature sensor indicates that the drive's temperature is exceeding the recommended temperature of the library.	See Error Code 1 in Drive Error Codes: Single-character display (SCD) .	Warning Event
37	Drive voltage	25h	Set when the drive detects that the externally supplied voltages are either approaching the specified voltage limits or are outside the voltage limits.	See Error Code 2 in Drive Error Codes: Single-character display (SCD) .	Warning Event
38	Predictive failure	26h	A hardware failure of the tape drive is predicted. Call the tape drive supplier helpline.	Predictive failure of drive hardware	Warning Event
39	Failure	27h	The tape drive might have a fault. Check for availability of diagnostic information and run extended diagnostics if applicable. Check the tape drive user's manual for instructions on running extended diagnostic tests and retrieving diagnostic data.	The drive might have a failure that can be identified by stored diagnostic information or by running extended diagnostics (Send Diagnostics).	Warning Event
49	Diminished Native Capacity	31h	Set when Native Capacity is diminished.	No action is required. Informational message only.	Info Event
51	Tape directory invalid at unload	33h	Set when the tape directory on the tape cartridge that was previously unloaded is corrupted. The file-search performance is degraded.	Use your backup software to rebuild the tape directory by reading all the data.	Warning Event
52	Tape system area write failure	34h	Set when the tape cartridge that was previously unloaded cannot write its system area successfully.	Copy the data to another tape cartridge, and discard the old cartridge.	Warning Event
53	Tape system area read failure	35h	Set when the tape system area cannot be read successfully at load time.	Copy the data to another tape cartridge, and discard the old cartridge.	Warning Event
55	Load Failure	37h	The operation failed because the media cannot be loaded and threaded.	Remove the tape and try another. If the problem persists, contact your IBM® service representative. Ensure that tape drive firmware is at the latest version. See Minimum firmware levels for common library features .	Warning Event
56	Unrecoverable unload failure	38h	The operation failed because the media cannot be unloaded.	Contact your IBM® service representative.	Warning Event
59	WORM Medium – integrity check failed	3Bh	Set when the drive determines that the data on tape is suspect from a WORM point of view.	1. Copy the data to another WORM tape cartridge. 2. Discard the faulty WORM tape.	Warning Event
60	WORM Medium – Overwrite attempted	3Ch	Set when the drive rejects a write operation because the rules for allowing WORM writes are not met. Data can be appended only to WORM media. Overwrites to WORM media are not allowed.	Write the data to a WORM tape cartridge or write the data to a non-WORM tape cartridge.	Warning Event

Sense data

When a drive encounters an error, it provides sense data as a response to the host.

Refer to the *IBM TS4300 Tape Library SCSI Reference* for library sense data information. Refer to the *IBM LTO Ultrium Tape Drive SCSI Reference* for tape drive sense data information.

In addition, you can use the IBM Tape Diagnostic Tool (ITDT) to further examine data and determine errors. See [IBM Tape Diagnostic tool \(ITDT\)](#).

Drive Error Codes: Single-character display (SCD)

SCD drive error codes give descriptions of the errors and messages that pertain to the drive.

If you encounter problems while the tape drive is running, refer to [How the library reports problems](#).

The SCD display appears on the inside back of an installed library and can be seen through the front window of an expansion unit. It is seen on the lower center of a full-height drive. Each drive has a status light and single-character display, that when visible provides drive informational and error conditions. The single-character display shows either a single character, a dot (in the lower right of the display), or both.

Note: The SCD can be seen through the windows of expansion units, and for full-height drives only. If you have a base unit without expansion units, or if you have half-height drives, the SCD cannot be seen.

[Single-character display \(SCD\) codes](#) lists the codes for error conditions and informational messages. If multiple errors occur, the code with the highest priority (represented by the lowest number) displays first. When the error is corrected, the code with the next highest priority displays until no errors remain.

The SCD is blank during normal operation.

Single-character display (SCD) codes

[Table 1](#) gives descriptions of the errors and messages that pertain to the drive. For troubleshooting tips, see [Troubleshooting Guide](#).

- Make note of the SCD error code before a cartridge is removed or the SCD error code is cleared.
- If an error occurred with a cartridge in the drive, eject the cartridge from the drive with the library Management GUI (see [Locating Management functions](#)).

Attention: If the drive detects a permanent error and displays an error code other than SCD **0**, it automatically runs a drive dump. If you force a drive dump, the existing dump is overwritten and data can be lost. After you force a drive dump, do not turn OFF the power to the drive or you might lose the dump data.

Table 1. Error codes on the single-character display

Error code	Meaning
0	No error occurred and no action is required. This code displays when diagnostics finish running and no error occurred. Note: The single-character display is blank during normal operation of the tape drive.
1	Temperature problem. The tape drive detected that the recommended operating temperature was exceeded.
2	Power problem. The tape drive detected that the externally supplied power is outside the specified voltage limits (the tape drive is not operating).
3	Firmware problem. The tape drive determined that a firmware error occurred Note: Do not force a new dump; the tape drive already created one.
4	Firmware or hardware problem. The tape drive determined that a firmware or tape drive hardware failure occurred. Note: Do not force a new dump; the tape drive already created one.
5	Tape drive hardware problem. The drive determined that a tape path or read/write error occurred. Notes: • To prevent damage to the drive or tape, the tape drive does not allow a cartridge to be inserted if the current cartridge was successfully ejected. • Do not force a new dump; the tape drive already created one.
6	Tape drive or media error. The tape drive determined that an error occurred, but it cannot isolate the error to faulty hardware or to the tape cartridge. Ensure that the tape cartridge is the correct media type. See LTO media . Ensure tape drive is at the latest firmware version. See Minimum firmware levels for common library features .
7	Tape drive or media error. The tape drive determined that an error occurred, but it cannot isolate the error to faulty hardware or to the tape cartridge. Ensure that the tape cartridge is the correct media type. See LTO media .
8	Interface problem. The tape drive determined that a failure occurred in the tape drive hardware or in the host bus. Note: The error code clears 10 seconds after the drive detected the error.
9	Tape drive or library-drive communication error. The tape drive determined that a failure occurred in the tape drive's hardware or in the library-drive connection.
A	Degraded operation. The tape drive determined that a problem occurred which degraded the operation of the tape drive, but it did not restrict continued use. If the problem persists, determine whether the problem is with the drive or the media. Note: The drive is usable, though the single-character display continues to indicate an error and the status light flashes amber.
C	The tape drive needs to be cleaned. See Locating Management functions .
d	Fiber AL_PA conflict. Two drives on fiber loop have the same AL_PA.
e	Encryption error. Displayed when the drive detects an error that is associated with an encryption operation.
E	Fiber Port offline. Displayed when the drive fiber port received a port bypass command from another port on the Fibre Channel network.
F	Fibre Channel error. No light is displayed if the drive fiber port does not detect light.
P	Write operation to a write protected cartridge was attempted. This action includes any attempt to overwrite a WORM protected tape. Ensure that the tape cartridge is the correct media type. See LTO media .
C	The drive is performing media optimization.

- [SCD dot](#)

- [Status light](#)

SCD dot

If a drive dump is present while the drive is in maintenance mode, a single red dot illuminates in the lower right corner of the SCD. To download the drive dump, see [Locating Management functions](#) or [IBM Tape Diagnostic tool \(ITDT\)](#).

The SCD dot turns OFF when you obtain a dump or update the drive firmware.

Note: If the drive dump is stored in ROM memory (SCD dot ON solid), the dump is lost when you turn OFF the power or reset the drive.

Status light

The Status light is a light-emitting diode (LED) that provides information about the state of the drive. The light can be green or amber, and (when lit) solid or flashing. [Table 1](#) lists the conditions of the Status light and single-character display (SCD) and provides an explanation of what each condition means.

Table 1. Meaning of Status light and single-character display (SCD)

If the Status light is...	And the SCD is...	Meaning
OFF	OFF	The drive has no power or is powered OFF.
Green	OFF	The drive is powered ON and in an idle state.
Flashing green	OFF	The drive is reading from the tape, writing to the tape, rewinding the tape, locating data on the tape, loading the tape, or unloading the tape.
Flashing green	OFF	The drive contains a cartridge during the power-ON cycle. In this case, the drive completes POST and slowly rewinds the tape (the process can take up to ten minutes). The light stops flashing and becomes solid when the drive completes the recovery and is ready for a read or write operation.
Flashing amber	Displaying error code	The drive is displaying error codes from the error code log on the SCD.
Amber	Red numbers, letters, or segments	During the power-on/initialization and POST (power-on self-test), the SCD briefly displays 8 , then becomes blank (not lit) when POST is complete and no POST errors occur. If a POST error is detected, an error code is displayed in the SCD and the Status light flashes amber.
Amber	Flashing 0	The drive is exiting from maintenance mode.
Amber	Flashing function	The drive is running the maintenance function.
Flashing amber once per second	Displaying error code	An error occurred and the drive or media might require service, or it might require cleaning.
Flashing amber once per second	Displaying C	The drive needs cleaning.
Flashing amber twice per second	OFF	The drive is updating firmware.
Flashing amber twice per second	OFF	The drive detected an error and is running a firmware recovery. It resets automatically.
Flashing amber twice per second	Flashing C	The drive is requesting a cartridge to be loaded.
Flashing amber twice per second	OFF	A drive dump is in flash memory.
Flashing green	Displaying C	The drive is performing media optimization.

Power must not be removed from the drive until the microcode update is complete. The drive indicates that the update is complete by resetting and running POST.

Upgrading and servicing

In this section, you can follow the procedures to add, remove, and replace library components.

Recommended tools

- #2 Phillips screwdriver
- Small Flat Head or Torx screwdriver

Identifying a failed component

Check which module contains the failed component. See [Identifying a failed component](#). If replacement parts are needed, go to [Replacement parts](#).

- [Internal view of library](#)
- [Adding, removing, or replacing a tape drive](#)
- [Adding or replacing a Base or Expansion Module](#)
- [Adding, removing, or replacing a power supply](#)
- [Replacing a Base or Expansion controller card](#)
- [Installing, removing, or replacing an accessor and spooling mechanism](#)
- [Removing or replacing a spooling mechanism](#)
- [Removing or replacing a magazine](#)
- [Moving the library modules](#)

Internal view of library

Figure 1. Internal view of the library

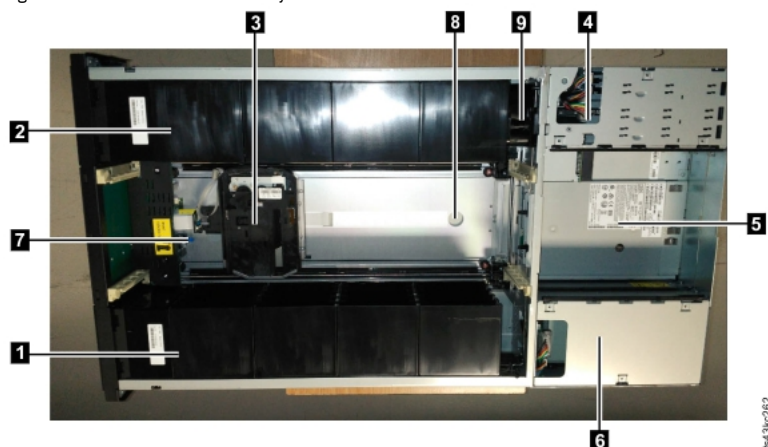


Table 1. Internal view description

Number	Item	Description
1	Right cartridge magazine	If the module is on the bottom, the right cartridge magazine holds 16 cartridges; if anywhere else in the library, it can hold 20 cartridges.
2	Left cartridge magazine	If the module is on the bottom, the left cartridge magazine holds 16 cartridges; if anywhere else in the library, it can hold 20 cartridges.
3	Accessor	This component contains the library accessor and bar code reader. The accessor moves cartridges to and from the • I/O station • Storage slots • Tape drive
4	Controller Card	This component is a customer replaceable unit (CRU) and stores the user configuration information or vital product data (VPD).
5	Tape drive	The module can contain a half-height or a full-height tape drive. The drive is a customer replaceable unit (CRU), and is designed for easy removal and replacement.
6	Power supply	The power supply is a customer replaceable unit (CRU) and the sole source of power for the module. The module is shipped with one power supply, but can contain an optional second power supply for redundancy.
7	Robotic lock lever	This component is used to lock down the accessor so it cannot move. This action is done when the module is moved or when the robotic assembly is removed.
8	Finger hole	One of two finger holes that are used to lift out the accessor assembly. The other is under the accessor in the photograph.
9	Spooling mechanism	This component moves the accessor.

Adding, removing, or replacing a tape drive

	CAUTION: Static Sensitive Risk of damage to devices • A discharge of static electricity damages static-sensitive devices or micro circuitry. • Proper packaging and grounding techniques are necessary precautions to prevent damage.

Table 1. Pinch hazard

	CAUTION: Risk of pinching hands or fingers. Can trap hands, fingers, and cause serious injury. Keep hands clear during operation. (L012)



Warning: Only individuals who are informed about the procedures and risks can replace or upgrade this tape drive assembly. Read all troubleshooting documentation and procedures before you proceed with repair or upgrade procedures. Hazardous moving parts exist inside this product. Do not insert tools or any portion of your body into the drive bay openings.

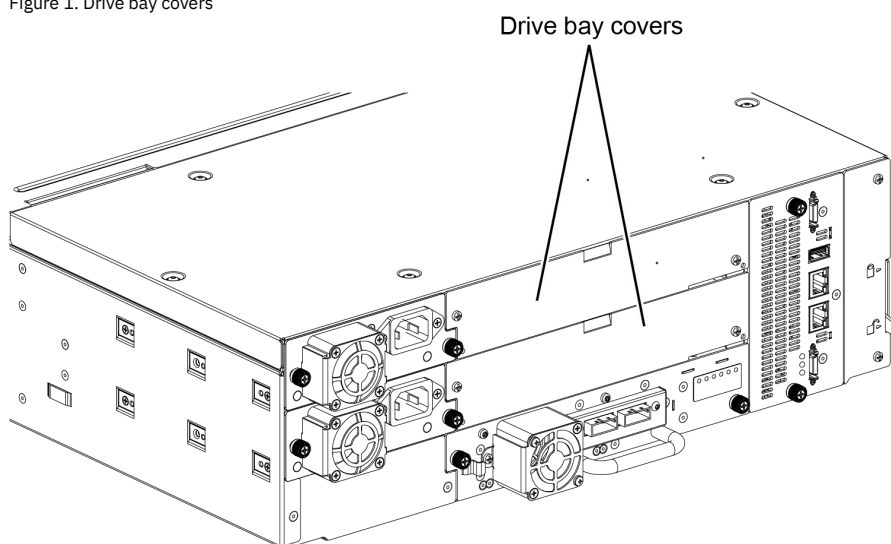
Important: ESD events occurring during tape drive installation or removal may cause SAS link reset on tape drives installed in the library. If this occurs, restart any jobs that were running on affected SAS links.

Adding a tape drive

Remember:

- Half-height tape drives can be installed in any drive bay in a module.
 - Full-height tape drives must be installed in the lowest two bays of a module. Installing a full-height drive in the top two bays of a module is not supported.
1. If you are adding a tape drive, remove a drive bay cover. With a Philips screwdriver, remove one half-height drive bay cover to install one half-height drive, or remove two drive bays covers to install a full-height tape drive.

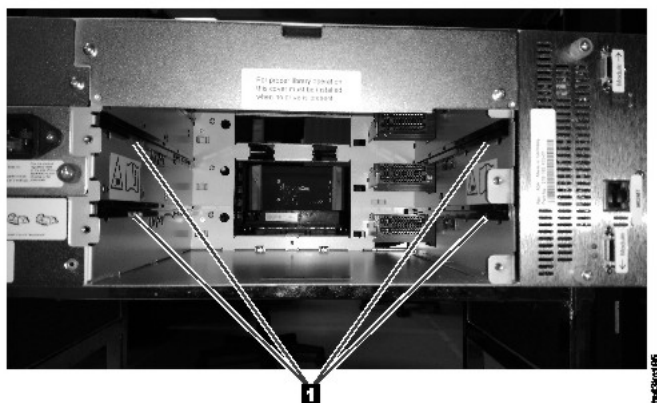
Figure 1. Drive bay covers



Note: A full-height tape drive must be installed in the lowest bay of the module.

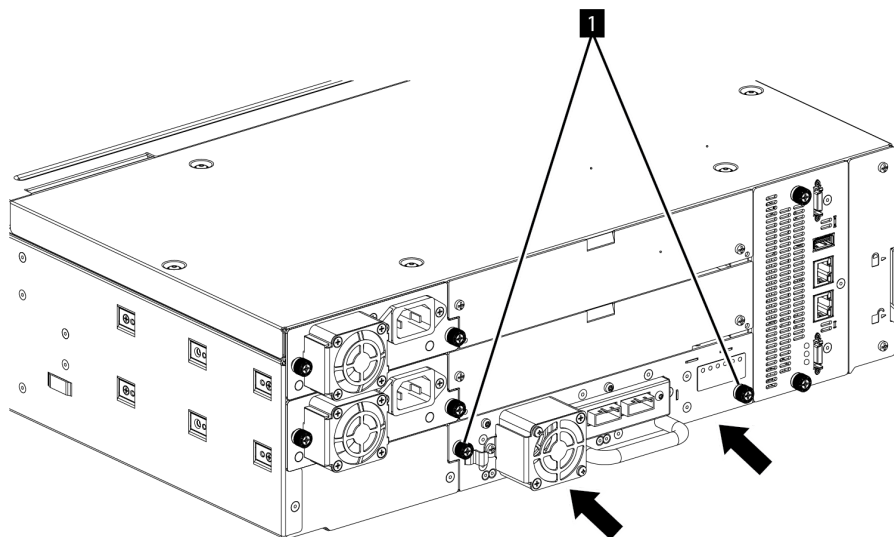
2. Align and slowly insert the new tape drive into the drive bay along the alignment rails (1 in [Figure 2](#)) while the drive assembly is supported. The tape drive must be flush with the back panel of the library.

Figure 2. Alignment rails



3. Tighten the captive thumbscrews (1 in [Figure 3](#)) with your fingers until the tape drive is secure.

Figure 3. Installing a tape drive

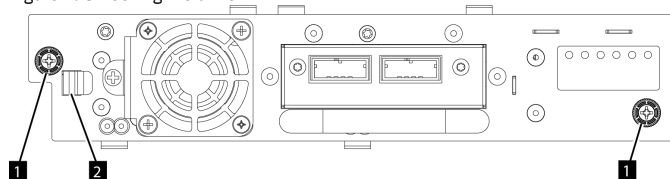


4. Verify the drive operation.
5. Use one of the logical library wizards to add the drive to a logical library as needed.

Removing a tape drive

- Ensure that all host activity, including library operations are stopped to the drive that is being removed. Exercise extra caution in case of a control path drive. Removing a control path drive will have severe impact on the operation of logical library the drive was installed in.
- Ensure that the tape cartridge is removed from the tape drive. Use the Management GUI to move the cartridge to a storage slot or I/O station.
- Remove the FC or SAS cables from the tape drive.
- Loosen the blue captive thumbscrews (1 in [Figure 4](#)) on the tape drive. Press the lock lever (2 [Figure 4](#)) to the right and pull straight back on the tape drive handle while the bottom of the drive is supported to remove it from the unit.

Figure 4. Unlocking the drive



Attention: Support the bottom of the tape drive when it is removed to avoid damaging any of the internal connections.

- Reset the list of known drives and modules. See [Locating Management functions](#).
- Confirm that the drive is logically removed by checking the Operator Panel or Management GUI.
- If replacement drive is not available, install the drive bay cover.
- If you are replacing the drive, see [Adding a tape drive](#).

Verifying Drive installation and operation

Using the Operator Panel or the Management GUI:

1. Confirm that the library recognizes the new tape drive by checking the Operator Panel or Management GUI. The new drive appears in the module status overview area on the left side of the screen.
2. Use the Management GUI or Operator Panel to verify that the tape drive has the current firmware. Update the firmware if necessary.
3. Use the Management GUI or Operator Panel to test the drive. See [Locating Management functions](#).

Adding or replacing a Base or Expansion Module

 18.1-33.6 kg (40-74 lbs)	Warning: Product Weight Caution: The weight of this part or unit is between 18.1 and 33.6 kg (40 and 74 lb). It takes two persons to safely lift this part or unit. (C009) Caution: The weight of this part or unit is between 33.6 and 46.3 kg (74 and 102 lb). It takes three persons to safely lift this part or unit. (C010) Risk of personal injury Before a module is moved or lifted • Observe local health and safety requirements and guidelines for manual material handling. • Remove all tapes to reduce the weight and to prevent cartridges from falling into the robotics path and damaging the library. • Remove all tape drives to reduce the weight. • Obtain adequate assistance to lift and stabilize the module during installation or removal.

CAUTION:  32-45 kg (70.5-121.2 lbs) or  >32 kg (70.5 lbs) or  >32 kg (70.5 lbs)	Risk of damage to devices When a module is placed into or removed from a rack • Extend the rack's leveling jacks to the floor. • Ensure that the full weight of the rack rests on the leveling jacks. • Install stabilizing feet on the rack. • Extend only one rack component at a time.
 >32 kg (70.5 lbs)	CAUTION: Parts can be damaged by electrostatic discharge. Keep parts in electrostatic containers until needed. Ensure that you are properly grounded when static sensitive components are touched.

Adding a module: Overview

To add a module to an existing configuration, you will

1. Power down the library.
2. Remove the top or bottom plate of the module. See [Preparing top and bottom modules](#).
3. Install the module into the rack. See [Installing modules in a rack](#).
4. Align and connect the modules. See [Aligning and connecting modules](#).
5. Connect the components and cables. See [Replacing the Module components and cables](#).
6. Connect the power cords, power on the library, and complete [Verifying Library Module installation and configuration](#).
7. Add tape cartridges to the new module.

Replacing a module: Overview

To replace the module, you will

1. Update the library firmware to the minimum code level of 1.2.1.0-A00 or higher.
2. Save the library configuration. See [Saving the configuration](#).
3. Remove tape cartridges and power off the library. See [Removing the magazines and cartridges](#) and [Powering off the library](#).
4. Remove all the components from the module and disconnect the power cords and cables. See [Removing the Module cables](#).
5. Remove the module from the rack. See [Removing the Module from a rack](#).
6. Install the replacement module into the rack. See [Installing the Module into a rack](#).
7. Replace the components and cables. See [Replacing the Module components and cables](#).
8. Connect the power cords, power on the library, and complete [Verifying Library Module installation and configuration](#).
9. Replace the tape cartridges.

You need a T-10 Torx screwdriver to remove the drive bay covers and a small flat head screwdriver. Have several static safe bags available for the boards that are moved to the replacement chassis.

Before the replacement procedure is begun

- Ensure that the rack is level side to side and front to back.
- Verify that any applications that are using the library are idle.

Attention: If the temperature in the room where the replacement module is installed varies by 15 C (59 F) from the room where it was stored, allow it to acclimate to the surrounding environment for at least 12 hours before it is unpacked from the shipping container.

Saving the configuration

See [Locating Management functions](#) for instructions on saving configuration settings to a file or an FAT32 formatted USB flash drive with the Management GUI or with the Operator Panel. This action is needed only for the Base chassis module and only as an extra safety precaution for both chassis and controller card replacement.

Note: Do not do a Save Configuration on a library that is in a failed state. Save the configuration on a working library only.

Removing the magazines and cartridges

For detailed instructions, see [Locating Management functions](#) to open the magazines.

Note: As a best practice, complete this procedure while applications are idle. While the magazine is pulled or removed, the library robotic assembly cannot move media.

Powering off the library

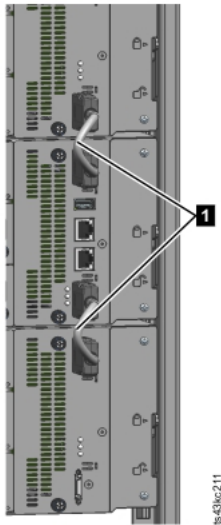
Power off the library from the front panel. Depress Power and hold it for 5 seconds. If the library does not complete a soft shutdown, depress and hold Power for 10 seconds.

Important: Under normal circumstances, when the library is powered off by using the front Power, the robot automatically parks and locks into the Base Module behind the Operator Panel. If you are given a choice during the power down procedure, choose the default park position. To protect the spooling cable or other sensitive parts, the accessor **must** be in the Base Module before any modules are removed from the library. If it is not, follow the procedure for returning the accessor to the base module. See [Returning the accessor to the Base Module](#).
Verify that all host processes are idle.

Removing the Module cables

1. Remove the power cords from the module that is replaced.
2. Remove the expansion interconnect cables (1) from the module that is replaced and from the modules that are connected to it.

Figure 1. Interconnect cables



Note: Completely removing the cables from both ends prevents damaging the expansion interconnect cables during module removal and replacement.

3. Remove any SAS, FC, or Ethernet cables from the module that is replaced.
4. Remove the USB device, if present.

Removing the tape drives

Remove any tape drives from the module that is replaced. The library tracks the drive locations and issues events if the drives aren't in the expected locations. Note the drive locations so they can be replaced in the same order and drive bays.

1. Use your fingers to loosen the blue captive thumbscrews on the tape drive.
2. Pull straight back on the tape drive handle while the bottom of the drive is supported to remove it from the module.

Attention: Support the bottom of the tape drive when it is removed to avoid damaging any of the internal connections.

Removing the power supplies

While the power supplies are removed, be sure to support the bottom. For detailed instructions, see [Adding, removing, or replacing a power supply](#).

Removing the Base or Expansion controller card

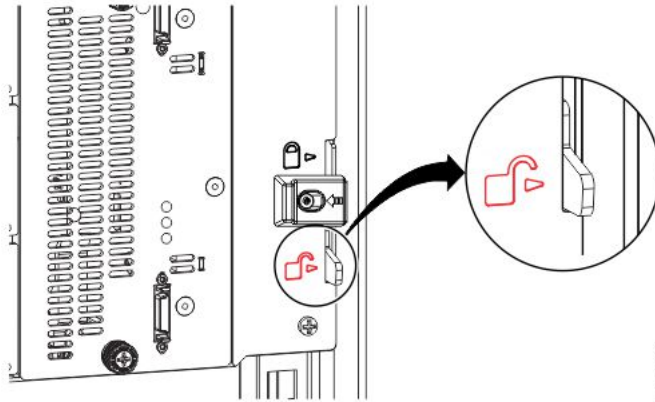
For detailed instructions, see [Replacing a Base or Expansion controller card](#).

Removing the Module from a rack

Obtain assistance to lift and stabilize the module during removal and replacement.

- If you are removing a module that has a module immediately above or below it,
 1. From the front of the library, use a #2 Phillips screwdriver to loosen the screws two full turns on the module and its adjacent modules.
 2. From the back of the library, unlock the alignment mechanisms that connect the module with the adjacent modules.
Note: If a blue alignment lever lock is attached to the rear of the module, slide it to the left, then move the alignment lever. The lever lock has an internal spring, so hold it while the alignment lever is moved, and it automatically springs back into place after the lever is moved. See [Figure 2](#).

Figure 2. Unlocking or disengaging the alignment lever



From the front of the library, use a #2 Phillips screwdriver and your fingers to loosen the captive thumbscrews screws two full turns on the module to be removed (circled in [Figure 3](#)). Then, slide the module out of the rack.

Figure 3. Loosening the thumbscrews

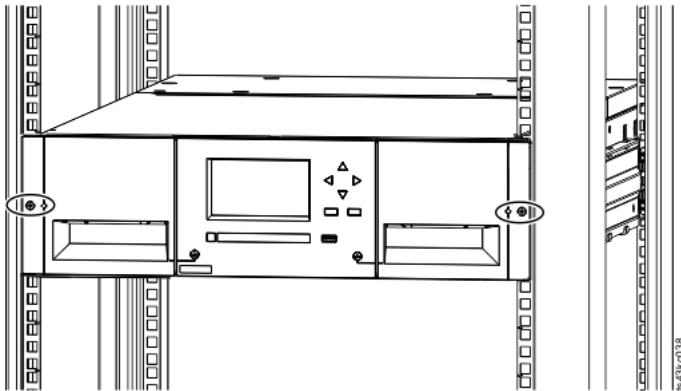
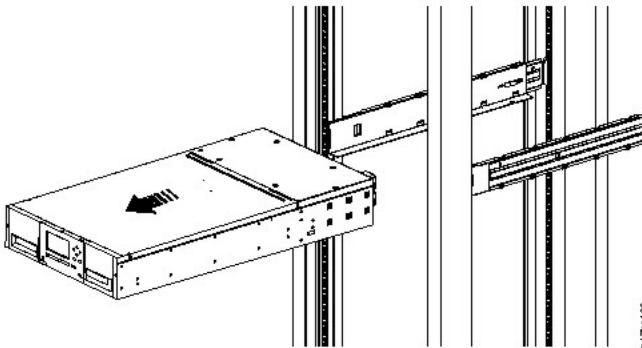


Figure 4. Sliding the module out of the rack



Moving the library cover plates

Unpack the replacement module and place it on a sturdy work surface. Save the packaging materials to return the empty module.

The Base Module has removable top and bottom cover plates. The two covers are identical and the process for removing and installing them is the same for the top and bottom of the module. See [Preparing top and bottom modules](#) for details. While this procedure refers to moving a cover from the Base Module, the information is the same for moving a cover from an Expansion Module. The covers must be removed only if the failed unit does not have the covers (the cover is on another module).

The replacement module is shipped with a bottom cover plate but not a top cover plate. Move the cover plates as necessary so the replacement module has the cover plates in the same location as the empty module and the empty module has a bottom cover plate.

Installing the Module into a rack

See [Installing modules in a rack](#) for details.

Replacing the Module components and cables

Replace the module components by reversing the removal procedures. Align the components carefully in the guide slots and tighten thumbscrews only with your fingers. If the thumbscrews cannot be tightened easily, verify that the component is aligned properly.

1. Replace the controller card. See [Replacing a Base or Expansion controller card](#).
2. Replace the tape drives in the same locations.

Tip: To help align the drive, remove the drive bay covers for one drive at a time.

See [Adding, removing, or replacing a tape drive](#).

3. Replace the magazines in the same locations.
4. Replace the power supplies. See [Adding, removing, or replacing a power supply](#).
5. Reattach any SAS, FC, expansion interconnect, and Ethernet cables that was removed earlier.
6. Reinsert the USB device if you removed it earlier.
7. Reattach the power cords.

Verifying Library Module installation and configuration

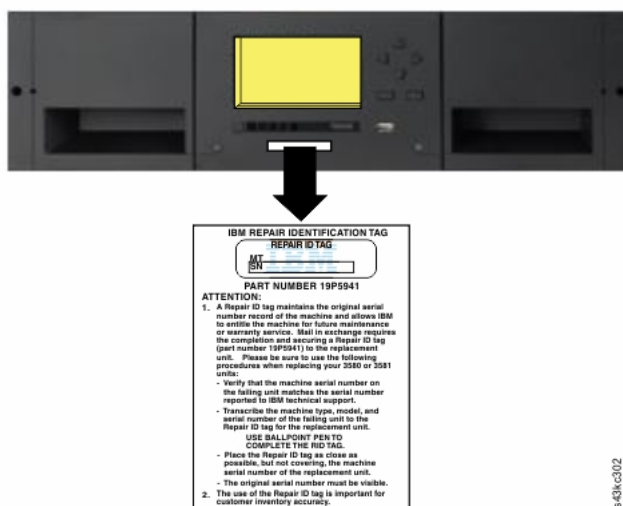
- Power on the library.
- Verify that the library initializes correctly and that the status is Ready.
- Run Library Verify to verify that the replacement module is visible in the Operator Panel or Management GUI.
- If a module was replaced, validate the library configuration in the Management GUI at Library > Logical Libraries.
- Properly complete the Repair Identification (RID) tag that was included with the replacement module.

Note: A RID (Repair Identification) tag maintains the original serial number record of the module to ensure your warranty coverage, if applicable, is not interrupted.

The tag is important for customer inventory accuracy. Follow the instructions on the RID tag precisely.

- Copy the serial number of the defective module onto the RID tag.
- Apply the tag to the front of the new replacement module.

Figure 5. Placement of the RID tag (Base Module shown)



- If a module was added, you must reset your logical libraries by using the basic logical library wizard. See [Locating Management functions](#) to find and run the basic logical library wizard.

Adding, removing, or replacing a power supply



CAUTION:
Static Sensitive

Risk of damage to devices

- A discharge of static electricity damages static-sensitive devices or micro circuitry.
- Proper packaging and grounding techniques are necessary precautions to prevent damage.

Important: ESD events occurring during power supply installation or removal may cause SAS link reset on tape drives installed in the library. If this occurs, restart any jobs that were running on affected SAS links.

Removing the power supply

Figure 1. Power supplies

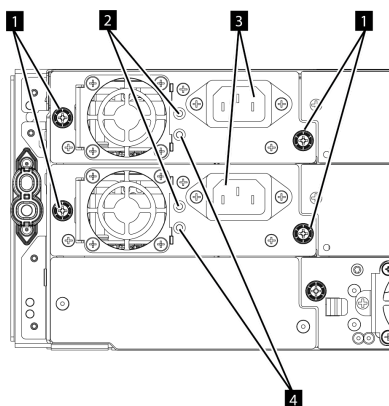


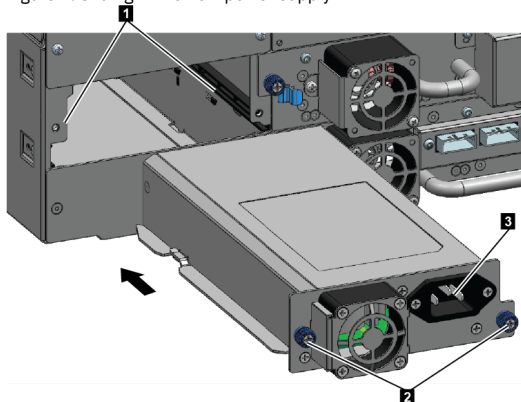
Table 1. Power supply components

1	Blue captive thumbscrews
2	White, lit if the AC power is connected.
3	AC power outlet
4	Green, lit if the module in turned on.

1. Locate the failed power supply on the rear of the library by the UID LEDs notification, and also by the power supply LEDs. Either the green LED (4) is lit or both LEDs are unlit.
2. Unplug the AC power cord (3) from the power supply you are replacing.
3. Loosen the two blue captive thumbscrews (1) with your fingers on the power supply.
4. Using the thumbscrews (one on each side), slowly pull the power supply approximately 10 cm (4 inches) from the back of the module.
5. Use one hand to completely remove the power supply from the module, while the other hand is used to support the bottom.

Adding or replacing the power supply

Figure 2. Sliding in the new power supply



1. Position the new power supply onto the alignment rails (1).
2. Slide the power supply into the module until it is flush with the back panel of the module.
3. Tighten the blue captive thumbscrews (2) with your fingers to secure it to the module.
4. Attach the AC power cord to the new power supply (3) and plug the power cord into an outlet.

Installing a secondary power supply

1. Position the secondary power supply onto the alignment rails. Note the positions of the primary versus the secondary power supplies - top versus middle bay.
2. Slide the power supply into the module until it is flush with the back panel of the module.
3. Tighten the blue captive thumbscrews with your fingers to secure it to the module.
4. Attach the AC power cord to the new secondary power supply.

Verifying the power supply installation and operation

1. Verify that the new power supply is operating properly by checking the power supply LEDs.
 - The white (2 in [Figure 1](#)) LED is lit.
 - The green (4 in [Figure 1](#)) LED is unlit.

Note: If a second power supply is installed when the TS4300 is powered on (concurrent MES), the green LED is lit.

With the Operator Panel or Management GUI, confirm that the power supply is operating correctly. The event that indicated that the power supply was faulty is cleared.
2. If the UID LEDs are still illuminated, deactivate them by using the Operator Panel or Management GUI.

Replacing a Base or Expansion controller card



CAUTION:

- Parts can be damaged by electrostatic discharge. Keep parts in electrostatic containers until needed. Ensure that you are properly grounded when static sensitive components are touched.
- You must power off the library to install or replace this part or damage can occur.

Important: Do not replace both the base chassis and the Base Module controller card with repair components in the same procedure. The firmware does not allow the library to operate if both components are replaced at the same time. Critical library information is saved in the controller card and within the chassis. When one is replaced, the data from the original component is transferred to the repair component. If both the base chassis and Base Module controller are replaced, you must power cycle the library between component replacements.

Saving the configuration

See [Locating Management functions](#) for instructions on saving configuration settings to a file with the Management GUI or with the Operator Panel. This procedure is done when a Base Module controller card is replaced, or as an extra precaution when both the controller card and the module are replaced.

Note: Do not do a Save Configuration on a library that is in a failed state. Save the configuration on a working library only.

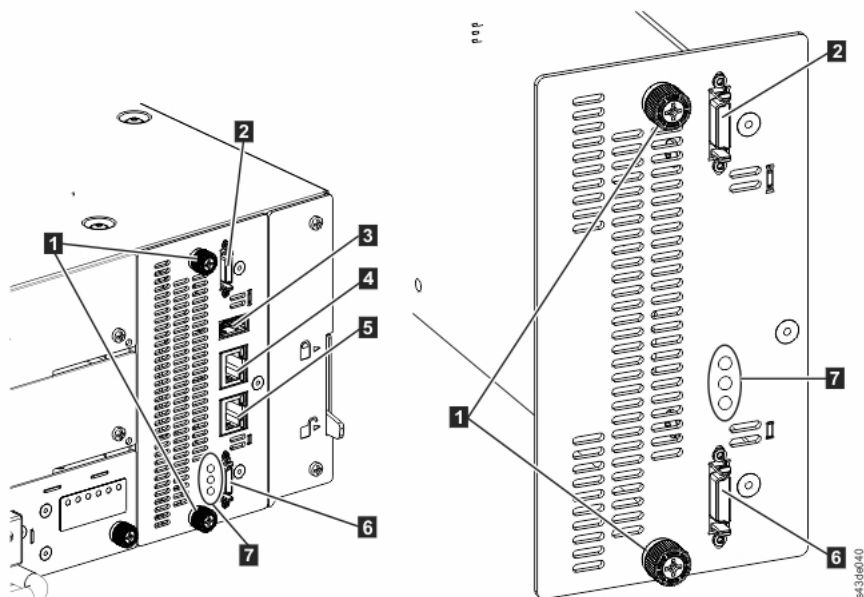
Powering off the library

Verify that all host processes are idle, then power off the library from the front panel. Depress Power and hold it for 5 seconds. If the library does not complete a soft shutdown, depress and hold Power for 10 seconds.

Important: Under normal circumstances, when the library is powered off by using the front Power, the robot automatically parks and locks into the Base Module behind the Operator Panel. If you are given a choice during the power down procedure, choose the default park position. To protect the spooling cable or other sensitive parts, the accessor **must** be in the Base Module before any modules or drives are removed from the library. If it is not, follow the procedure for returning the accessor to the base module. See [Returning the accessor to the Base Module](#).

Removing the controller card

Figure 1. Controller card components



Note: The base controller card is on the left, and the expansion controller card is on the right.

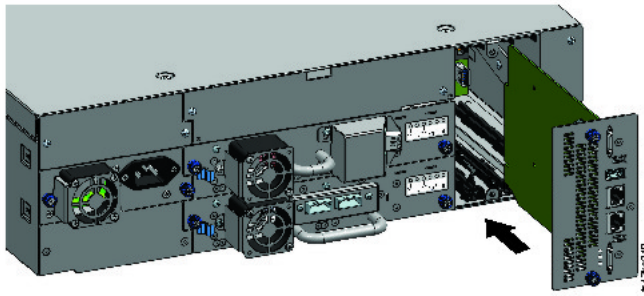
Table 1. Controller card components

1	Blue captive thumbscrews
2	Upper Expansion Module connection port
3	USB Port
4	Ethernet Port A
5	Ethernet Port B
6	Lower Expansion Module connection port
7	Controller card LEDs, top to bottom • Green Controller Health Status. The flashing LED indicates that the controller is in good health status and properly working. • Yellow Controller Error. This LED turns on if the controller has a hardware issue. In this case, the green LED stops flashing. • Blue Unit Identifier. This LED is a beacon that can be turned on or off through the Management GUI. The LED gives the user an indication that the controller needs attention. See Identifying a failed component.

1. Unplug the AC power cables from the module that contains the failed controller card.
2. On the module that contains the failed controller card, remove the expansion interconnect cables (2 and 6) that connect to other modules, if present.
3. Remove the Ethernet cables (4 and 5) and the USB cable (3), if present. (An Expansion Module does not have Ethernet or USB ports. See [Figure 1](#)).
4. Loosen the two blue captive thumbscrews (1) on the controller.
5. Using the thumbscrews, slowly remove the controller from the module.

Installing the Base or Expansion controller card

Figure 2. Installing a Controller card



Important: Base and Expansion Module controller cards are keyed to fit in their respective modules only. A Base Module controller card **does not fit** into an Expansion Module, and vice versa. If you encounter resistance when the controller card is installed, make sure that you are installing the controller card into the appropriate module.

1. Position the new controller card on the alignment rails.
2. Slide the controller card slowly into the module until it is flush with the back panel of the module.
3. Tighten the blue captive thumbscrews (1) with your fingers to secure it to the module.
4. Replace the expansion interconnect cables (2 and 6), the Ethernet cable or cables (4 and 5), and the USB cable (3) that were removed previously. (An Expansion Module does not have Ethernet or USB ports. See [Figure 1](#)).
5. Plug in the AC power cables.

Powering on the library

Power on the library by pressing Power on the Base Module just below the Operator Panel. The green light illuminates. When the library is powered on, it inventories the tape cartridges in the magazines, checks the firmware version on all modules, and configures the tape drives. Then, the library confirms the presence of the existing modules, and searches for any new modules.

Verifying the Base or Expansion controller card

1. Verify that the library has the most up-to-date firmware revision. To find the version of firmware that is installed on the library, check the Library Properties page of the Management GUI or the Status > Library page of the Operator Panel.
2. If the Base Module controller is replaced, upgrade the firmware if necessary. Update the firmware from the Management GUI at Library > Actions > Update Library Firmware.
Important: If you are asked whether to retain the serial number, always select Yes.
3. Check the Attention light on the front panel and login to web interface and check the dashboard for any alerts.
4. With the Operator Panel or the Management GUI, check for any events. The event that indicated that the controller was faulty is cleared.
5. If the base module controller is replaced, the library configuration is automatically restored. Validate the library configuration, and complete a Restore if the library configuration was not restored.
6. If the UID LEDs are still illuminated, deactivate them by using the Operator Panel or Management GUI.
7. Resume the host applications.

Installing, removing, or replacing an accessor and spooling mechanism



CAUTION:

Parts can be damaged by electrostatic discharge. Keep parts in electrostatic containers until needed. Ensure that you are properly grounded when static sensitive components are touched.

Powering off the library

Verify that all host processes are idle, then power off the library from the front panel. Depress Power and hold it for 5 seconds. If the library does not complete a soft shutdown, depress and hold Power for 10 seconds.

Important: Under normal circumstances, when the library is powered off by using the front Power, the robot automatically parks and locks into the Base Module behind the Operator Panel. If you are given a choice during the power down procedure, choose the default park position. To protect the spooling cable or other sensitive parts, the accessor **must** be in the Base Module before any components are removed from the library. If it is not, follow the procedure for returning the accessor to the base module. See [Returning the accessor to the Base Module](#).

Preparing to remove the accessor and spooling mechanism from the Base Module



Warning:

When a module is extended from the library - to reduce the risk of personal injury or damage to equipment

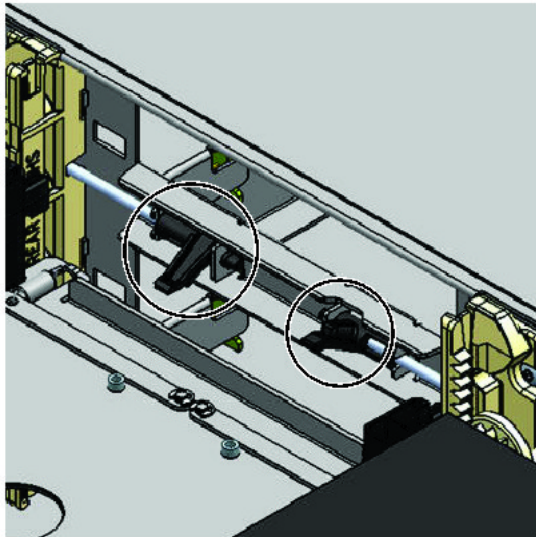
- Extend the rack-leveling jacks to the floor.
- Ensure that the full weight of the rack rests on the leveling jacks.
- Verify that the rack is level side to side and front to back.
- Install the rack stabilizer kit on the rack.
- Extend only one rack component at a time. Racks can become unstable if more than one component is extended.

1. Loosen the front captive screws that connect the Base Module to the rack two full turns.
2. If adjacent Expansion Modules exist
 - a. Loosen the front captive screws two full turns on the adjacent expansion modules.
 - b. Unlock the alignment lever.
 - c. Disconnect and completely remove the expansion interconnect cables from the Base Module and from the adjacent modules. Removing the expansion interconnect cables completely prevents damaging the cables when the module is moved in and out of the rack.
3. Disconnect the power supply cables on the Base Module.
4. Disconnect the Ethernet, SAS, and Fibre Channel cables from the Base Module.
5. Completely loosen the front captive screws of the Base Module.
6. Slowly extend the Base Module from the front of the rack and remove it from the rack.
7. Place the Base Module on a flat, level surface, such as a table.
8. Remove the top library cover plate, if present.
 - a. Unlock the top cover with two small screwdrivers.
 - b. Remove the cover from the module.

Removing the accessor and spooling mechanism from the Base Module

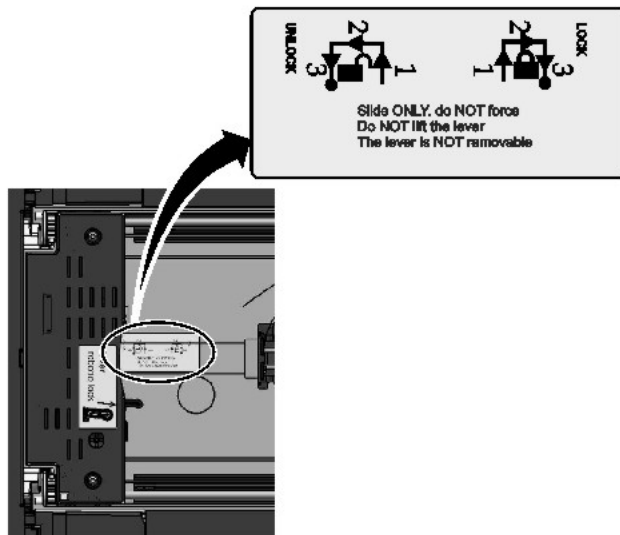
1. Remove the left and right magazines by using the magazine release levers (circled in [Figure 1](#)). Push up on the lever, then pull the magazine out.

Figure 1. Magazine release levers



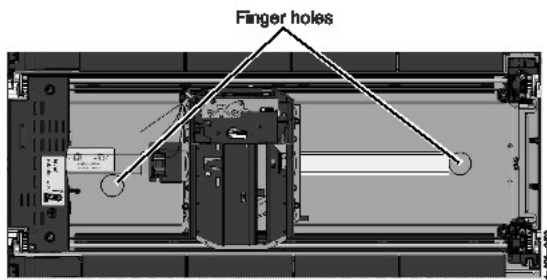
2. Slide the cartridge carrier toward the center of the accessor to access the robot-locking lever.
3. Standing at the front of the module, unlock the robot by moving the blue lever to the left, then toward you, then to the right.

Figure 2. Unlocking the robot



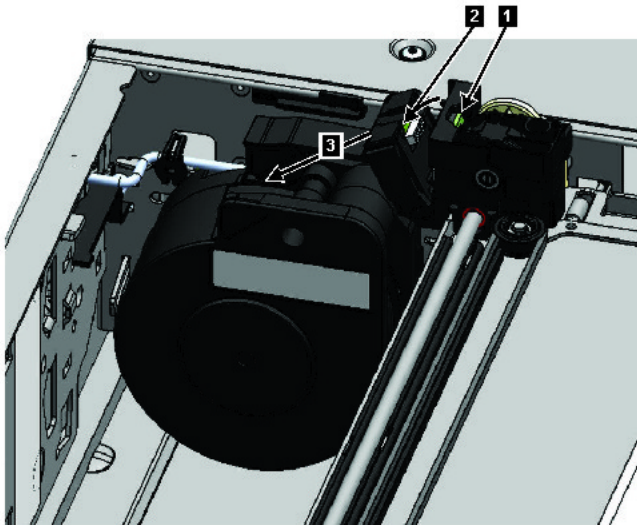
4. Place your fingers into the large holes on the accessor and pull up slowly.
Note: The accessor offers resistance. Lift the accessor no faster than 12 mm (0.5 inches) per second.

Figure 3. Finger holes



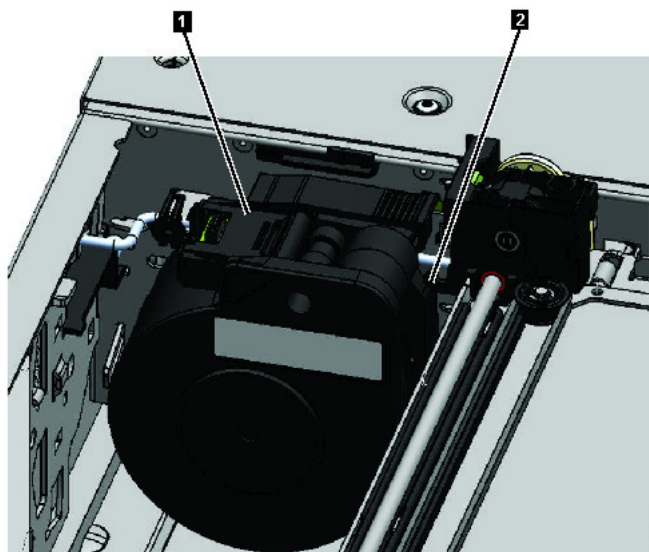
5. Lift the accessor gently from the module and place it on top of the gear mechanism. Take care not to damage the spooling cable.
6. Lock the robot to keep it from lowering
7. On the top of the accessor where the spooling cable is attached, press down on the latch (1 in [Figure 4](#)), then tilt out the piece that holds the spooling cable (2).
 Note: Note where the end of the spooling cable pivots in the accessor. It is important to know when you attach the new spooling cable to the accessor. See 2 in [Figure 5](#).
8. Lift the spooling cable from the accessor and place it in its cradle at the top of the spooling mechanism (3 in [Figure 4](#)).

Figure 4. Unlocking the spooling cable and placing it in its cradle



9. Place the spooling connector (1 in [Figure 5](#)) to the park position.

Figure 5. Spooling cable in park position



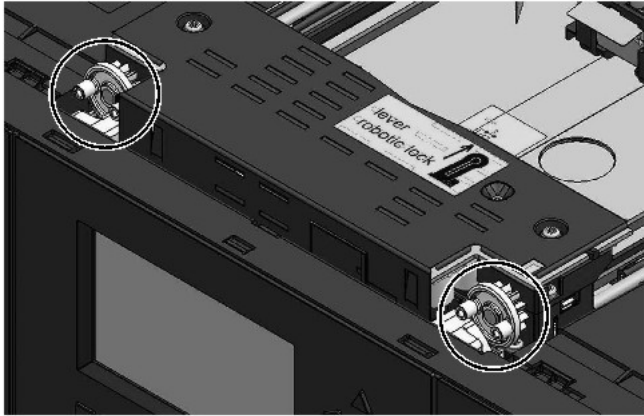
10. Unlock the lever and set aside the accessor. See [Figure 2](#).
 Important: If a tape cartridge is still in the cartridge carrier, remove the cartridge by lifting it straight up. You might need to move the cartridge slightly from side to side.
11. Replace the spooling mechanism. Refer to [Removing or replacing a spooling mechanism](#).

Installing the accessor into the Base Module

Important: If an accessor assembly is replaced, the minimum library firmware must be 1.2.1.0-A00. Go to Library > Actions > Update Library Firmware to update the library firmware.

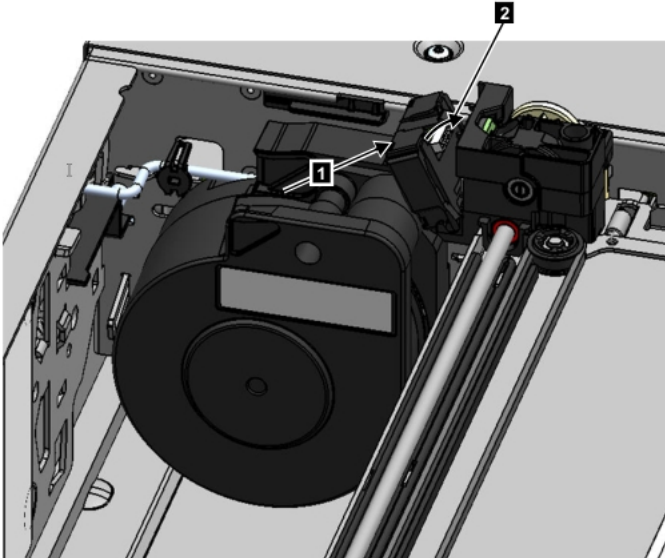
1. Each corner of the accessor has a gear with two protruding pins. Rotate one of the gears on the accessor so that the two pins are aligned horizontally. See [Figure 6](#)

Figure 6. Pins are aligned horizontally



2. The accessor is shipped with the robot in the unlocked position. Verify that the replacement unit is **locked**, then set it on top of the gears.
3. Place the gears of the accessor into the grooves on the inside corners of the module. **Confirm that all four of the pins are touching the outside of the grooves.**
4. Standing at the right side of the module, remove the end of the spooling cable from the park position.
5. Place the spooling cable into the grooves where it attaches to the accessor and rotate it until it snaps into place. See [Figure 7](#).

Figure 7. Installing the spooling cable



6. Unlock the accessor. The accessor drops smoothly. If it does not, check the alignment of the gears.
7. Before the accessor gets to the bottom, lock the robot. Standing at the front of the module, move the blue lever to the left, then away from you, then to the right.
Tip: If the end of the spooling cable drops into the module, unlock the accessor, remove it from the module, return the end of the spooling cable to its cradle, return the accessor to its previous position in the module, relock the accessor, and repeat the procedure.

After the accessor and spooling mechanism installation

1. Push the magazines back into the module until they lock into place.
2. Replace the top cover on the Base Module if you removed one.
3. Slide the module into the rack.
4. If no adjacent modules exist, tighten the front screws.
5. If adjacent modules exist
 - a. Set the alignment mechanisms to the lock position. If you encounter resistance, adjust the upper module so the pin in the alignment mechanism moves into the hole in the lower module.
 - b. When the alignment mechanism is in the locked position, tighten the front screws on the module
 - c. Reconnect the expansion interconnect cables.
6. Reconnect the Ethernet, SAS, and Fibre Channel cables to the Base Module.
7. Reconnect the power supply cables to the Base Module.
8. Pack the failed accessor and spooling mechanism to return to your service.

Powering on the library

1. Power on the library by pressing Power on the Base Module just below the Operator Panel for 5 seconds.

2. The green light illuminates.
3. When the library is powered on, it inventories the tape cartridges in the magazines, checks the firmware version on all modules, configures the tape drives, confirms the presence of the existing modules, and searches for any new modules.

Verifying the installation

1. Verify that the library powers on and initializes correctly, and that the status is Ready.
2. If the UID LEDs are still illuminated, deactivate them by using the Operator Panel or Management GUI.
3. Run Library Verify to ensure that the library is working correctly.

- [Returning the accessor to the Base Module](#)

Returning the accessor to the Base Module

If you powered off the library and the accessor did not return to its park position in the Base Module behind the Operator Panel.

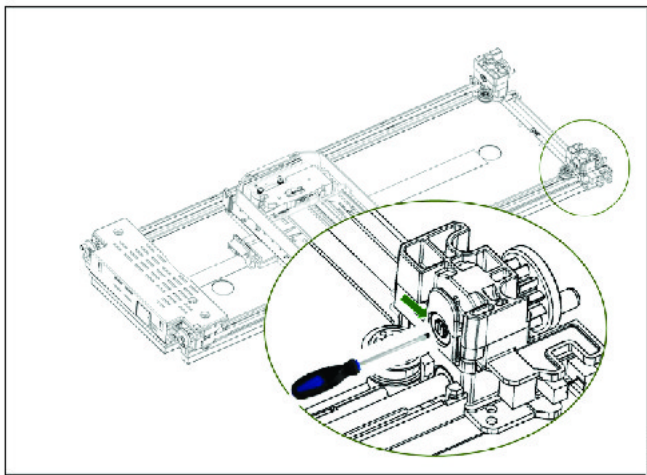
1. Power on the library by pressing Power on the Base Module just below the Operator Panel.
2. Return the accessor to its park position.
3. Power off the library by pressing Power on the Base Module and holding for 5 seconds.

If the accessor is still not in the Base Module, use one of the procedures in the following two sections.

If the accessor is stopped in an Expansion Module that is near the Base Module or is stopped directly between two modules.

1. Remove the front bezel from the Base Module, the Expansion Module containing the accessor, and modules in between as needed.
2. Insert a small flat head screwdriver into the screwdriver relief on the right rear bearing block of the accessor.

Figure 1. Inserting the screwdriver to manually operate the accessor

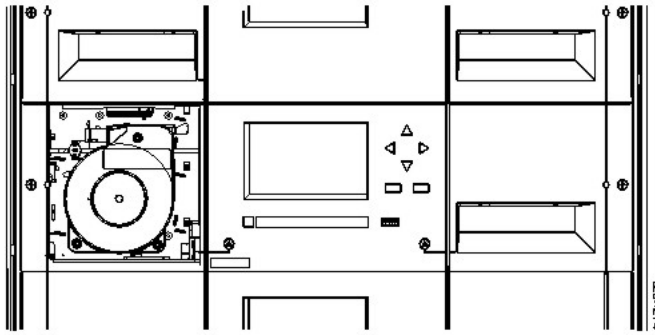


3. Turn the screwdriver to manually operate the accessor gear train and move the accessor into the Base Module.
4. Lock the accessor. Standing at the front of the module, move the blue lever to the left, then away from you, then to the right.
5. Reinstall the bezels that were previously removed.
6. Remove the accessor and spooling mechanism. See [Preparing to remove the accessor and spooling mechanism from the Base Module](#).
7. Install the new accessor and spooling mechanism. See [Installing the accessor into the Base Module](#).
8. Slide the Base Module back into the rack. See [After the accessor and spooling mechanism installation](#).

If the accessor is stopped in an Expansion Module that is not near the Base Module or it cannot move vertically.

1. Remove the left magazine of the Base Module. See [Removing or replacing a magazine](#). The library must already be powered off. Therefore, you must unlock the magazine by using the manual release.
2. Disconnect the power supply cables from all of the modules.
3. Using plastic-handled scissors, reach through the left magazine opening of the Base Module and carefully cut the spooling cable.

Figure 2. Left magazine opening



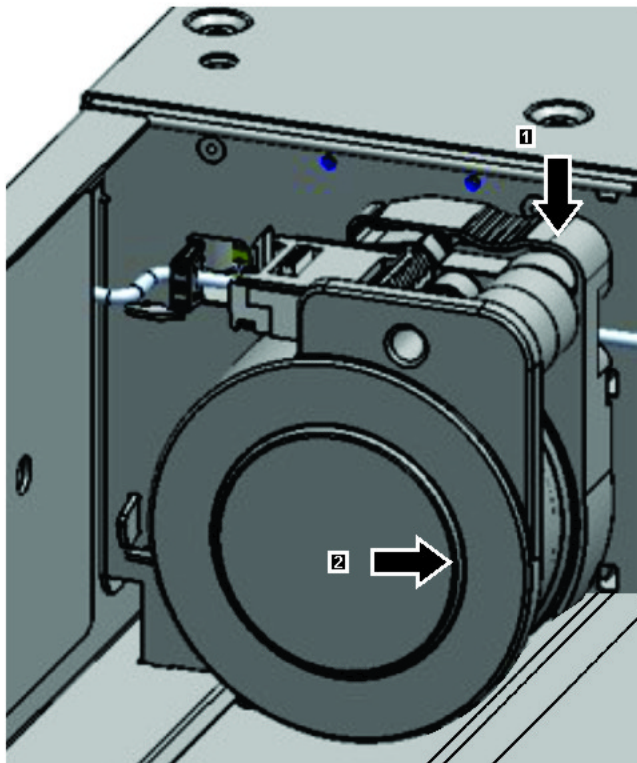
4. Extend the expansion module that contains the accessor while carefully guiding the free spooling cable. See [Preparing to remove the accessor and spooling mechanism from the Base Module](#). While minor differences might occur, these instructions for a Base Module also apply to an Expansion Module.
5. Remove the accessor from the Expansion Module by using Step 1 through Step 7 in [Removing the accessor and spooling mechanism from the Base Module](#).
6. Slide the Expansion Module back into the rack. See [After the accessor and spooling mechanism installation](#). While minor differences might occur, these instructions for a Base Module also apply to an Expansion Module.
7. Extend the Base Module. See [Preparing to remove the accessor and spooling mechanism from the Base Module](#).
8. Remove the spooling mechanism from the Base Module by using Step 8 through Step 10 in [Removing the accessor and spooling mechanism from the Base Module](#).
9. Install the new accessor and spooling mechanism. See [Installing the accessor into the Base Module](#).
10. Slide the Base Module back into the rack. See [After the accessor and spooling mechanism installation](#).

Removing or replacing a spooling mechanism

Occasionally, only the robotic spooling cable must be removed and replaced. See [Identifying a failed component](#).

1. Power down the library.
Important: Under normal circumstances, when the library is powered off by using the front Power, the robot automatically parks and locks into the Base Module behind the Operator Panel. If you are given a choice during the power down procedure, choose the default park position. To protect the spooling cable or other sensitive parts, the accessor **must** be in the Base Module before any components are removed from the library. If it is not, follow the procedure for returning the accessor to the base module. See [Returning the accessor to the Base Module](#).
2. Refer to [Preparing to remove the accessor and spooling mechanism from the Base Module](#) for the steps in preparing your library.
3. Remove the left magazine to provide clear access to the spooling mechanism.
4. Follow the steps in [Removing the accessor and spooling mechanism from the Base Module](#) to remove the accessor, disconnect the spooling cable, and place it in the park position.
5. Push down on the lever on the top of the spooling mechanism (1) and slide about 10 mm towards the center (2) to unlock the mechanism.

Figure 1. Unlocking the spooling mechanism



Note: Make sure that the spooling mechanism is unlocked before you try to pull it out. When the mechanism is unlocked, the cutout behind the mechanism is covered, and the mechanism cannot slide any further to the center.

Figure 2. Unlocked spooling mechanism - enlarged view

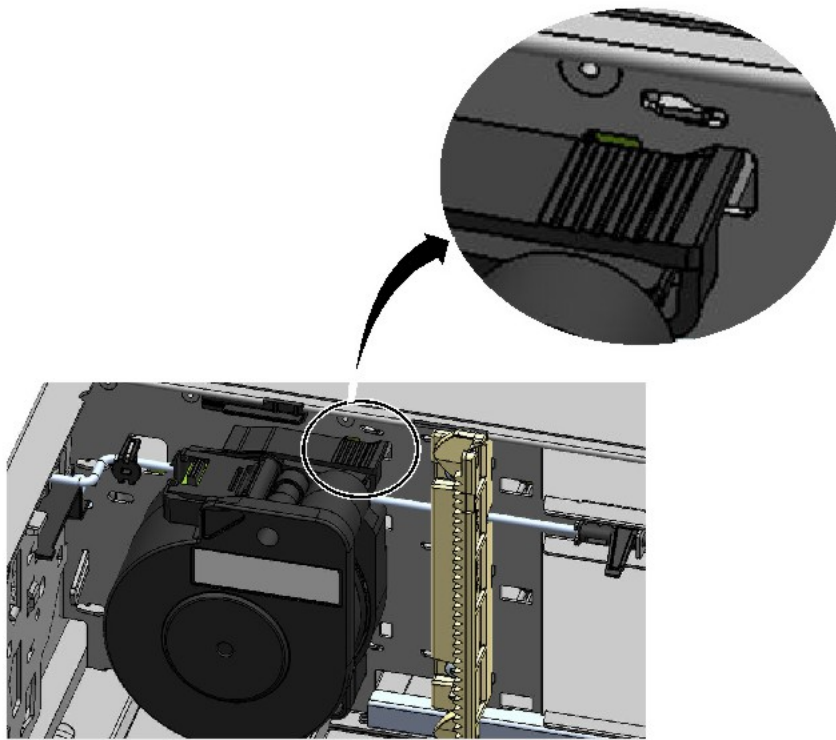
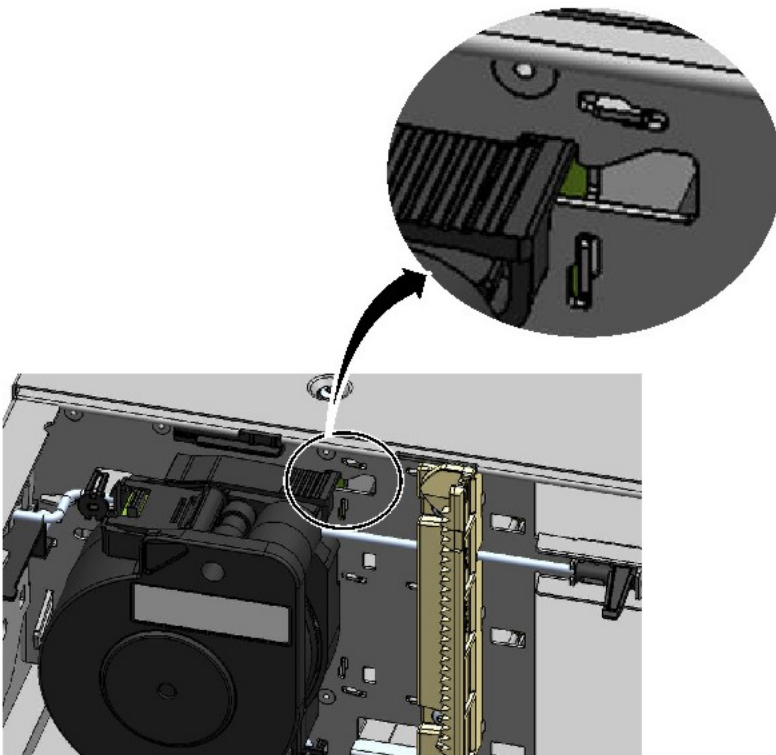
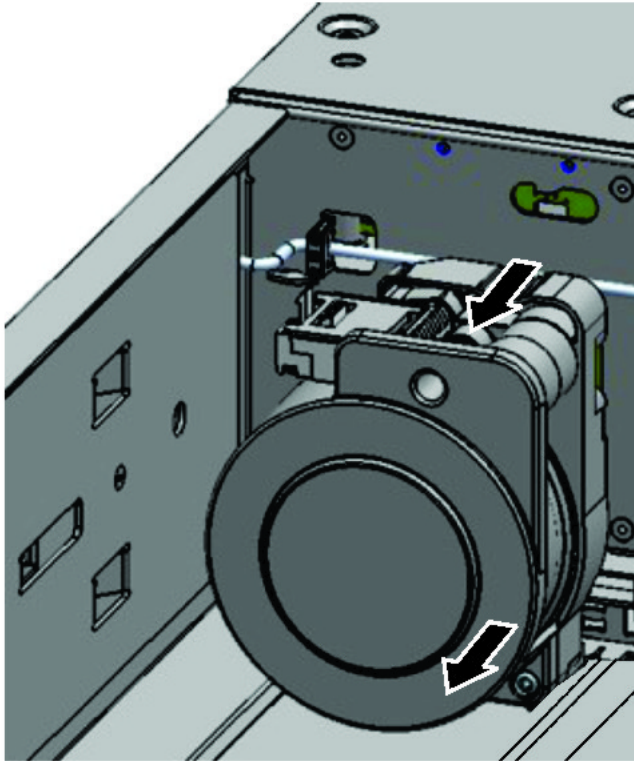


Figure 3. Locked spooling mechanism - enlarged view



6. Pull the spooling mechanism towards the front of the module to remove it.

Figure 4. Removing the spooling mechanism



7. Reverse the steps to replace the failed spooling mechanism with the new unit.
8. Follow the steps in [Installing the accessor into the Base Module](#) and [After the accessor and spooling mechanism installation](#) to put the library back into service.

Removing or replacing a magazine

It is recommended that you unlock the magazine with the Operator Panel, Management GUI, or the release button on the front panel. If these methods fail, or if a magazine needs to be removed when the power to the device is off, you can release the magazine manually. Only one magazine or I/O station can be open at a time. Note: This procedure is completed more effectively while applications are idle. While the magazine is extended, the library robotic assembly cannot move media.

1. Log in as an administrator.
2. Go to Library > Modules and Magazines.
3. On the Actions menu, click Unlock Magazine.
4. Click Open in the left or right magazine column within the module that contains the magazine to be opened.
5. A message box indicates when the magazine is unlocked.
6. Unlock Magazine screen shows that the magazine is now unlocked.

Note: If not removed, the magazine and the I/O station relock after 30 seconds.

To manually eject the magazine, insert a paper clip or a small flat head screwdriver into the appropriate magazine release hole and gently push the tab in. See [Figure 1](#) and [Figure 2](#).

1. Open the magazine access door.
2. Insert a paper clip or a small flat head screwdriver into the appropriate magazine release hole and gently push the tab in.

Figure 1. Manually releasing the right magazine



Figure 2. Manually releasing the left magazine



3. Pull the magazines out of the module.

Moving the library modules

When a library module is moved within the rack, to a different rack, or in a rack to a different physical location, care must be taken to avoid personal injury and damage to the module.

 18.1-33.6 kg (40-74 lbs) CAUTION:  32-55 kg (70.5-121.2 lbs)  OR  >32 kg (70.5 lb) OR  >32 kg (70.5 lb)	Warning: Product Weight Caution: The weight of this part or unit is between 18.1 and 33.6 kg (40 and 74 lb). It takes two persons to safely lift this part or unit. (C009) Caution: The weight of this part or unit is between 33.6 and 46.3 kg (74 and 102 lb). It takes three persons to safely lift this part or unit. (C010) Risk of personal injury Before a module is moved or lifted: • Observe local health and safety requirements and guidelines for manual material handling. • Remove all tapes to reduce the weight and to prevent cartridges from falling into the robotics path and damaging the library. • Remove all tape drives to reduce the weight. • Obtain adequate assistance to lift and stabilize the module during installation or removal. Risk of damage to devices When a module is placed into or removed from a rack: • Extend the rack's leveling jacks to the floor. • Ensure that the full weight of the rack rests on the leveling jacks. • Install stabilizing feet on the rack. • Extend only one rack component at a time.

Important: Under normal circumstances, when the library is powered off by using the front Power, the robot automatically parks and locks into the Base Module behind the Operator Panel. If you are given a choice during the power down procedure, choose the default park position. To protect the spooling cable or other sensitive parts, the accessor must be in the Base Module before any drives are removed from the library.

To move a module within a rack or into a different rack:

1. Save the library configuration.
2. Remove the tape cartridges from the tape drives and magazines, and power off the library.
3. Disconnect the power cords and cables, and unlock the alignment mechanisms.
Attention: Failure to disconnect all cables can result to damage to the cable or the mating electronic assembly in the library.
4. Remove the modules from the rack.
5. Remove the rack rails from the rack.
6. Verify that the destination rack is level side to side and front to back.
7. Install the rack rails in the destination rack.
8. Install the modules in the rack.
9. Replace the cables and lock the alignment mechanisms.
10. Connect the power cords, power on the library, and verify the operation.
11. Replace the tape cartridges.

For instructions for these steps, see [Adding or replacing a Base or Expansion Module](#) and [Installing](#).

Reference

This section provides information about the Help pages, REST API, LTO Media, and other reference documentation.

- [Minimum firmware levels for common library features](#)
- [Management GUI functions and roles](#)

- [TS4300 Help pages](#)
The IBM® Help pages provide an interactive view of the interfaces and their associated help pages.
- [REST API for scalable tape libraries](#)
The REST API is a simple application programming interface (API) to manage the 3U scalable tape libraries remotely over an HTTPS interface. This API is requested and needed for manufacturing and for automated test and monitoring systems.
- [Library Configuration Forms](#)
Make a copy of these forms, and fill them out as you are installing and configuring your library.
- [LTO media](#)
This section provides an overview of LTO media, which is recommended for use with this library.
- [Replacement parts](#)
TS4300 tape library has Tier 1 and Tier 2 CRUs (customer replaceable units). These CRUs are parts of the library that must be added, removed, and replaced by the customer. Tier 1 CRUs do not require tools for installation while Tier 2 CRUs need tools for installation.
- [Manual cartridge removal procedure](#)
Follow this procedure if a cartridge must be manually removed and repaired.
- [Accessibility](#)
Accessibility features help a user with a physical disability, such as restricted mobility or limited vision, to use the HTML version of the customer documentation successfully.

Minimum firmware levels for common library features

Table 1. Minimum firmware levels for common library features

Feature	Minimum Firmware Levels Required
LTO9 (HH/FH) Tape Drives	Library Firmware must be at 1.5.0.0-A00 or greater to support the LTO9 tape drives. Ensure that any host applications and device drivers are at the minimum level that is required to support LTO9 tape drives.
LTO8 (HH/FH) Tape Drives	Library Firmware must be at 1.1.1.0-A00 or greater to support the LTO8 tape drives. Ensure that any host applications and device drivers are at the minimum level that is required to support LTO8 tape drives.
LTO6 (HH/FH) and LTO7 (HH/FH) Tape Drives	Library Firmware must be at 1.1.0.1-A00 or greater to support the LTO6 and LTO7 tape drives. Ensure that any host applications and device drivers are at the minimum level that is required to support LTO6 and LTO7 tape drives.
Library-Managed Encryption	Library Firmware must be at 1.1.1.0-A00 or greater to support the Library-Managed Encryption feature. Ensure that any key manager applications are at the minimum level that is required to support the 3U library. SKLM/GKLM for z/OS encryption requires minimum library firmware 1.2.0.0-B00.
Path Failover (Control Path and Data Path)	Library Firmware must be at 1.1.1.0-A00 or greater to support the Path Failover feature. Ensure that any IBM® device drivers are at the minimum level that is required to support the 3U library.
Remote Logging (rsyslog)	Library Firmware must be at 1.1.1.0-A00 or greater to support the Remote Logging feature. Ensure that any IBM device drivers are at the minimum level that is required to support the 3U library.
LTO M8 media	Library Firmware must be at 1.1.1.1-B00 or greater to support the M8 media feature. Drive firmware must be at HB82 or greater to support the M8 media feature. Ensure that any IBM device drivers are at the minimum level that is required to support the 3U library.
Sequential Mode	Library Firmware must be at 1.1.1.2-A00 or greater to support Sequential Mode. Ensure that any host applications are at the minimum level that is required to support the 3U library in this mode.
Library Serial Number 7800K0K or greater	Library firmware must be at 1.2.1.0-A00 or greater to support newer library serial numbers.
Key Path Diagnostics (KPD)	Library Firmware must be at 1.3.0.0-A00 or greater to support KPD.

Management GUI functions and roles

The administrator can access all functions of the library and can make changes. Other user roles have restrictions on what features can be accessed or changed. An administrator can give others access to the library but can restrict their full capability. See [Managing](#) for an overview of the four user roles.

Table 1. Management GUI functions and roles

Management GUI Functions				Roles			
				Admin-istrator	Service	Super-user	Monitor
Library				✓	✓	✓	✓
	Dashboard			✓	✓	✓	✓
		Actions		✓	✓	✓	✓
			Inventory Library	✓	✓	✓	
			Update Library Firmware	✓	✓	✓	
			Export Library Logs	✓	✓	✓	
			Reset Library	✓	✓	✓	
			Turn Identifier Light On or Off	✓	✓	✓	
			Tests	✓	✓	✓	
			Library Verify	✓	✓	✓	
			Demo Mode	✓	✓	✓	
			Drive Test	✓	✓	✓	
			Slot to Slot Exerciser (0-10 cycles)	✓	✓	✓	✓
			Slot to Slot Exerciser (0-endless)		✓		
			Properties	✓	✓	✓	✓

Management GUI Functions				Roles			
				Admin-istrator	Service	Super-user	Monitor
	Modules and Magazines			✓	✓	✓	✓
		Actions		✓	✓	✓	
			Unlock I/O Station	✓	✓	✓	
			Unlock Magazine	✓	✓	✓	
			Enable or Disable I/O Station	✓	✓	✓	
		Modules		✓	✓	✓	✓
		Refresh		✓	✓	✓	✓
	Logical Libraries			✓	✓	✓	✓
		Actions		✓	✓	✓	✓
			Manage Logical Library (Expert Mode)	✓	✓	✓	
			Manage Logical Library (Basic Mode)	✓	✓	✓	
			Manage KMIP Encryption	✓	✓	✓	
			Manage SKLM for z/OS Encryption	✓	✓	✓	
			Graphical View	✓	✓	✓	✓
		Refresh		✓	✓	✓	✓
		Collapse All		✓	✓	✓	✓
	Events			✓	✓	✓	
		Error and Warning Events		✓	✓	✓	
			Actions	✓	✓	✓	
			Mark All Open Events Inactive	✓	✓	✓	
			Clear Log	✓	✓	✓	
			Include Inactive Events	✓	✓	✓	
		Service Events			✓		
			Actions		✓		
			Clear Log		✓		
		Informational Events		✓	✓	✓	
			Actions	✓	✓	✓	
			Clear Log	✓	✓	✓	
		Configuration Events		✓	✓	✓	
			Actions	✓	✓	✓	
			Clear Log	✓	✓	✓	
		Show All		✓	✓	✓	
Drive				✓	✓	✓	✓
	Drives and Ports			✓	✓	✓	✓
		Actions		✓	✓	✓	
			Modify Port Settings	✓	✓	✓	
			Clean Drive	✓	✓	✓	
			Reset Drive	✓	✓	✓	
			Eject Cartridge from Drive	✓	✓	✓	
			Drive Test	✓	✓	✓	
			Update Drive Firmware	✓	✓	✓	
			Export Service Logs	✓	✓	✓	
		Refresh		✓	✓	✓	✓
		Expand All		✓	✓	✓	✓
Cartridges				✓	✓	✓	✓
	Cartridges and Slots			✓	✓	✓	✓
		Actions		✓	✓	✓	✓
			Inventory Library	✓	✓	✓	
			Move Cartridges	✓	✓	✓	
			Graphical View	✓	✓	✓	
		Search Bar		✓	✓	✓	✓
		Clear		✓	✓	✓	✓
Access				✓	✓		
	Local User			✓			
		Add User		✓			
		Actions		✓	✓		
			Modify User Password (must click user)	✓			
			Modify Role Permissions	✓	✓		
			Modifv Operator Panel PIN	✓			

Management GUI Functions				Roles			
				Admin-istrator	Service	Super-user	Monitor
			Remove User (must click user)	✓			
			Filter by name	✓	✓		
	Local Password Policies			✓	✓		
			Password Rules	✓	✓		
			Submit	✓	✓		
	LDAP Authentication			✓	✓		
			LDAP Servers	✓	✓		
			Add Server	✓			
			Actions	✓			
			Modify Server (click server)	✓			
			Remove Server (click server)	✓			
			LDAP Users	✓	✓		
			Add User	✓			
			Actions	✓			
			Modify User	✓			
			Remove User	✓			
			LDAP User Groups	✓	✓		
			Add User Group	✓			
			Actions	✓			
			Modify User Group (click group)	✓			
			Remove User Group (click group)	✓			
	Kerberos Authentication			✓	✓		
			Kerberos Servers	✓	✓		
			Add Server	✓			
			Actions	✓			
			Modify Server (click Server)	✓			
			Remove Server (click Server)	✓			
			Kerberos Users	✓	✓		
			Add User	✓			
			Actions	✓			
			Modify User	✓			
			Remove User	✓			
Settings				✓	✓	✓	✓
	Library			✓	✓	✓	
			Date and Time	✓	✓	✓	
			Time Zone	✓	✓	✓	
			Date Time Format	✓	✓	✓	
			Set Date Time	✓	✓	✓	
			SNTP	✓	✓	✓	
			Licensed Features	✓	✓	✓	
			Add License Key	✓	✓	✓	
			Licensed Key in Systems	✓	✓	✓	
			Firmware Update	✓	✓	✓	
			Firmware Level	✓	✓	✓	
			Advanced	✓	✓	✓	
			Save Configuration File	✓	✓	✓	
			Restore Configuration File	✓	✓	✓	
			Reset Configuration Only	✓	✓	✓	
			Full Factory Reset	✓	✓		
			Reset the list of known Drives and Modules	✓	✓	✓	
			Initial Configuration Wizard	✓	✓	✓	
			Configuration Wizard Application	✓	✓	✓	
			Auto Calibration	✓	✓		
			Start Auto Calibration Wizard	✓	✓		
	Network			✓	✓	✓	

Management GUI Functions					Roles			
					Admin-istrator	Service	Super-user	Monitor
		Ethernet			✓	✓	✓	
			General Network Settings		✓	✓	✓	
			Primary Network Port		✓	✓	✓	
			IPv4		✓	✓	✓	
			Secondary Network Port		✓	✓	✓	
			Reset Internal IP Range (in case of conflict)		✓	✓	✓	
	Notifications				✓	✓	✓	
		Email			✓	✓	✓	
			Email SMTP Settings		✓	✓	✓	
		SNMP			✓	✓	✓	
			SNMP Settings		✓	✓	✓	
		Remote Logging (rsyslog)			✓	✓	✓	
			Remote Logging		✓	✓	✓	
	Security				✓	✓	✓	✓
		Encryption			✓	✓	✓	✓
			Actions		✓	✓	✓	
				Manage KMIP Encryption	✓	✓	✓	
				Manage SKLM for z/OS Encryption	✓	✓	✓	
				Run Key Path Diagnostics	✓	✓	✓	
			Security Encryption Status		✓	✓	✓	✓
			Refresh		✓	✓	✓	✓
		GUI			✓	✓	✓	
			Secure Communications		✓	✓	✓	
			Certificate Settings		✓	✓	✓	
			Create Custom Certificate		✓	✓	✓	
			Backup Custom Certificate		✓	✓	✓	
			Restore Custom Certificate		✓	✓	✓	
			Session Timeout		✓	✓	✓	
			Operator Panel/RMI Session Locking		✓	✓	✓	

TS4300 Help pages

The IBM® Help pages provide an interactive view of the interfaces and their associated help pages.

To download and access the IBM Help pages locally, right-click [IBM Help](#), click Save Link As.. or Save Target As.. (depending on your browser) to save the file to your system. Extract the file, and click index.html to start the interactive demonstration of the TS4300 helps.

Notes:

- You can use <http://www.7-zip.org/download.html> to download a free program to extract the files.
- Microsoft Edge and Internet Explorer browsers are supported for viewing the Help pages. Other browsers might not be supported.

REST API for scalable tape libraries

The REST API is a simple application programming interface (API) to manage the 3U scalable tape libraries remotely over an HTTPS interface. This API is requested and needed for manufacturing and for automated test and monitoring systems.

To download and access the IBM® REST API locally, right-click [REST API](#), click Save Link As.. or Save Target As.. (depending on your browser) to save the file to your system.

Library Configuration Forms

Make a copy of these forms, and fill them out as you are installing and configuring your library.

Update the forms each time that changes are made to the library configuration and store these forms in a secure location. Having the information on these forms are helpful if a call to IBM® service is necessary.

You can also save library configuration data from the Management GUI. See [Locating Management functions](#).

- [Library information](#)

- [Module and drive information](#)
- [Logical Library information](#)

Library information

General Information	
Library type	TS4300 (MT 3555)
Serial Number	
Host name	
SNTP server	
Encryption License Key	
Path Failover License Key	
Network Settings	
Domain Name	
Network Protocol	IPv4/IPv6
Max. Link Speed	
Method	
IP Address	
Gateway	
DNS1	
DNS2	
Security Settings	
Password Policy	
SSL	
Certificates	
LDAP Server (See also User Accounts)	
LDAP Domain	
Encryption Settings	
Encryption Key Manager Server 1/Port	
Encryption Key Manager Server 2/Port	
Notification Settings	
SMTP Notification Level	
SMTP Server/Port	
SMTP Security	
SMTP Email Address	
SNMP Community Name	
SNMP Notification Level	
SNMP Server/Port 1	
SNMP Server/Port 2	

Module and drive information

Make a copy of this page, for more than 2 modules.

Number	
Number of Power Supplies	
I/O Station Enabled	
Drive 1 (bottom slot) Type	
-- Serial Number	
-- Logical Library Number/Control Path	
-- Port Settings (FC only)	
Drive 2 (middle slot) Type	
-- Serial Number	
-- Logical Library Number/Control Path	
-- Port Settings (FC only)	
Drive 3 (top slot) Type	
-- Serial Number	
-- Logical Library Number/Control Path	
-- Port Settings (FC only)	
Number	
Number of Power Supplies	
I/O Station Enabled	
Drive 1 (bottom slot) Type	
-- Serial Number	
-- Logical Library Number/Control Path	

-- Port Settings (FC only)	
Drive 2 (middle slot) Type	
-- Serial Number	
-- Logical Library Number/Control Path	
-- Port Settings (FC only)	
Drive 3 (top slot) Type	
-- Serial Number	
-- Logical Library Number/Control Path	
-- Port Settings (FC only)	

Logical Library information

Make a copy of this page, for more than 2 logical libraries.

Number	
Name	
Number of Drives	
Number of Slots	
Number of I/O Slots	
Barcode Label Length Rep to Host	
Barcode Label Alignment Rep to Host	
Auto Clean	
Key Manager Type	
LTO7+ Multi-Initiator SCSI Conflict Detection	
Sequential Mode	
Number	
Name	
Number of Drives	
Number of Slots	
Number of I/O Slots	
Barcode Label Length Rep to Host	
Barcode Label Alignment Rep to Host	
Auto Clean	
Key Manager Type	
LTO7+ Multi-Initiator SCSI Conflict Detection	
Sequential Mode	

LTO media

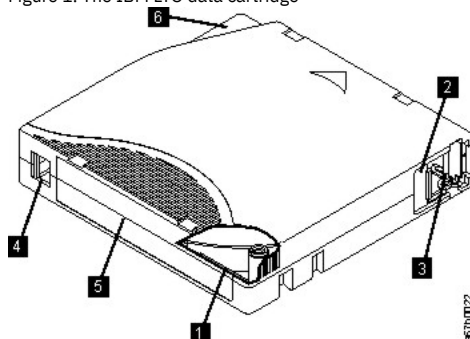
This section provides an overview of LTO media, which is recommended for use with this library.

To ensure that your IBM® tape drive conforms to IBM's specifications for reliability, use only IBM LTO tape cartridges. You might use other LTO-certified data cartridges, but they might not meet the standards of reliability that are established by IBM. The IBM LTO data cartridge cannot be interchanged with the media used in other IBM non-LTO tape products.

[Figure 1](#) shows the IBM LTO data cartridge and its components.

1	LTO cartridge memory	4	Write-protect Switch
2	Cartridge door	5	Label area
3	Leader pin	6	Insertion guide

Figure 1. The IBM LTO data cartridge



Note: The same components are on all the IBM LTO data cartridges.

- [Data cartridges](#)
The different generations of IBM Ultrium data cartridges can be identified by color.

- [WORM \(Write once, read many\) cartridges](#)
Certain records retention and data security applications require a write once, read many (WORM) method for storing data on tape.
- [Cleaning cartridge](#)
An IBM Ultrium Universal Cleaning Cartridge is required to clean the tape drive.
- [Cartridge memory chip \(LTO-CM\)](#)
All generations of the IBM LTO Ultrium data cartridges include a Linear Tape-Open Cartridge Memory (LTO-CM) chip.
- [Bar code labels](#)
Bar code label requirements for LTO drives.
- [Write-Protect switch](#)
The position of the write-protect switch on the tape cartridge determines whether you can write to the tape.
- [Cartridge care and handling](#)
Incorrect handling or an incorrect environment can damage cartridges or their magnetic tape.
- [Repositioning or reattaching a leader pin](#)
Information about repairing your tape cartridges: repositioning or reattaching the leader pins.
- [Environmental and shipping specifications for tape cartridges](#)
Specific storage and shipping environmental conditions apply to tape cartridges.
- [Disposing of tape cartridges](#)
Under the current rules of the US Environmental Protection Agency (EPA), regulation 40CFR261, the LTO Ultrium Tape Cartridge is classified as non-hazardous waste.
- [Ordering media supplies](#)

Data cartridges

The different generations of IBM® Ultrium data cartridges can be identified by color.

Table 1. Cartridge types and colors

Type	Color
Ultrium 9	Teal
Ultrium 9 WORM	Teal and Silver gray
Ultrium 8	Burgundy
Ultrium 8 WORM	Silvery gray
Ultrium M8	Purple
Ultrium 7	Purple
Ultrium 7 WORM	Purple and Silvery gray
Ultrium 6	Black
Ultrium 6 WORM	Black and Silvery gray
Ultrium 5	Burgundy
Ultrium 5 WORM	Burgundy and Silvery gray
Ultrium 4	Green
Ultrium 4 WORM	Green and Silvery gray
Ultrium 3	Slate Blue
Ultrium 3 WORM	Slate Blue and Silvery gray
Ultrium 2	Purple
Ultrium 1	Black

All generations contain 1/2-inch, dual-coat, magnetic tape.

You can order tape cartridges with the bar code labels included, or you can order custom labels. To obtain tape cartridges and bar code labels, see [Ordering media supplies](#).

When tape is processed in the cartridges, Ultrium tape drives use a linear, serpentine recording format. The native data capacity and recording format of Ultrium data cartridges is as follows:

Table 2. Cartridge data capacity and recording formats

Type	Native Data Capacity	Recording Format
Ultrium 9	18 TB (45 TB at 2.5:1 compression)	Reads and writes data on 8960 tracks, 32 tracks at a time.
Ultrium 8	12 TB (30 TB at 2.5:1 compression)	Reads and writes data on 6656 tracks, 32 tracks at a time.
Ultrium M8	9 TB (22.5 TB at 2.5:1 compression) ¹	Reads and writes data on 3584 tracks, 32 tracks at a time.
Ultrium 7	6 TB (15 TB at 2.5:1 compression)	Reads and writes data on 3584 tracks, 32 tracks at a time.
Ultrium 6	2.5 TB (6.25 TB at 2.5:1 compression)	Reads and writes data on 2176 tracks, 16 tracks at a time.
Ultrium 5	1.5 TB (3 TB at 2:1 compression)	Reads and writes data on 1280 tracks, 16 tracks at a time.
Ultrium 4	800 GB (1.6 TB at 2:1 compression)	Reads and writes data on 896 tracks, 16 tracks at a time.
Ultrium 3	400 [®] GB (800 GB at 2:1 compression)	Reads and writes data on 704 tracks, 16 tracks at a time.
Ultrium 2	200 GB (400 GB at 2:1 compression)	Reads and writes data on 512 tracks, 8 tracks at a time.
Ultrium 1	100 GB (200 GB at 2:1 compression)	Reads and writes data on 384 tracks, 8 tracks at a time.

¹Library Firmware must be at 0080 or greater to support the LTO M8 media feature. Drive firmware must be at HB82 or greater to support the LTO M8 media feature. Ensure that any IBM device drivers are at the minimum level that is required to support the library.

The first set of tracks is written from near the beginning of the tape almost to the end of the tape. The head then repositions to the next set of tracks for the return pass. This process continues until all tracks are written and the cartridge is full, or until all data is written.

The cartridge door (2 in [Figure 1](#)) protects the tape from contamination when the cartridge is out of the drive. The tape is attached to a leader pin (3 in [Figure 1](#)) behind the door. When the cartridge is inserted into the drive, a threading mechanism pulls the pin (and tape) out of the cartridge, across the drive head, and onto a non-removable take-up reel. The head can then read or write data from or to the tape.

The write-protect switch (4 in [Figure 1](#)) prevents data from being written to the tape cartridge. For more information, see [Write-Protect switch](#).

The label area (5 in [Figure 1](#)) provides a location to place a label. For more information, see [Bar code labels](#).

The insertion guide (6 in [Figure 1](#)) is a large, notched area that prevents the cartridge from being inserted incorrectly.

Table 3. Nominal cartridge life:
Load/unload cycles

Type	Load/Unload Cycles
Ultrium 9	100,000 (100k)
Ultrium 8	20,000 (20k)
Ultrium M8	20,000 (20k)
Ultrium 7	20,000 (20k)
Ultrium 6	20,000 (20k)
Ultrium 5	20,000 (20k)
Ultrium 4	20,000 (20k)
Ultrium 3	20,000 (20k)
Ultrium 2	10,000 (10k)
Ultrium 1	5000 (5k)

- [Cartridge Read/Write compatibility](#)
- [LTO type M cartridge \(M8\)](#)
- [Capacity scaling](#)

Cartridge Read/Write compatibility

Table 1. Data cartridge compatibility with LTO tape drive

IBM® LTO Tape Drive	IBM LTO Data Cartridges						
	LTO 9	LTO 8	LTO M8	LTO 7	LTO 6	LTO 5	LTO 4
LTO8 tape drive		Read/Write	Read/Write	Read/Write			
LTO7 tape drive				Read/Write	Read/Write	Read only	
LTO6 tape drive					Read/Write	Read/Write	Read only

LTO type M cartridge (M8)

The LTO Program introduced a new capability with LTO-8 tape drives: the ability to write 9 TB (native) on a brand new LTO Ultrium 7 cartridge instead of 6 TB (native) as specified by the LTO-7 format. Such a cartridge is called an LTO7 initialized LTO Type M cartridge. These LTO Type M cartridges are identifiable by using an automation bar code label that ends with the last 2 characters “M8”.

Table 1. LTO7 and LTO8 Cartridge Types

Cartridge/Density type	Bar code label	Cartridge Packaging/Silkscreen labeling	Native capacity	Tape Drive compatibility
L8	xxxxxxL8	LTO Ultrium 8	12 TB	LTO8
M8	xxxxxxM8	LTO Ultrium 7	9 TB	LTO8
L7	xxxxxxL7	LTO Ultrium 7	6 TB	LTO7, LTO8

From now on, these cartridges are referred to as L8, M8, and L7.

Only new, unused LTO Ultrium 7 cartridges can be initialized as M8 cartridges. When a cartridge is initialized as M8, it cannot be changed back to L7. Initialized M8 cartridges can be written and read-only in an LTO8 tape drive. LTO7 tape drives cannot read initialized M8 cartridges.

M8 cartridges can be purchased as either pre-initialized (also referred to as “labeled and initialized”) M8 data cartridges or uninitialized M8 data cartridges (M8 WORM cartridges are not supported). For either option, the bar code label is included. However, the uninitialized M8 data cartridge must first be initialized in tape libraries that support the automatic initialization of uninitialized M8 cartridges while under the control of ISV applications that recognize the “M8” bar code label.

A tape cartridge is initialized when it is first loaded into a compatible tape drive and data is written by the ISV application at the beginning of tape (sometimes referred to as “labeling a tape” or “writing from BOT”). The tape drive then establishes the density of the media.

If an uninitialized M8 cartridge is not initialized in a tape library that supports uninitialized M8 cartridges, then the cartridge might inadvertently and silently be initialized at the L7 density (that is, at a 6 TB native capacity) even if the bar code label states “M8”. This action might occur with the usage of non-TS2900 Tape Autoloaders, stand-alone LTO-7 tape drives, stand-alone LTO-8 tape drives, earlier LTO-8 tape drive firmware, earlier TS2900 Tape Autoloader firmware, or earlier ISV software that does not recognize that M8 cartridges must be mounted only in LTO8 tape drives. M8 cartridges that are inadvertently initialized at the L7 density can continue to be read and written in LTO-7 and LTO-8 tape drives. However, they remain limited to the 6 TB native capacity.

TS2900 Tape Autoloader firmware version 0080 added support for uninitialized M8 cartridges, in addition to support for pre-initialized M8 cartridges. In any tape product with M8 cartridges, the minimum LTO8 tape drive firmware version is HB82.

Capacity scaling

To control the capacity of the cartridge (for example, to obtain a faster seek time) issue the SCSI command **SET CAPACITY**. For information about this command, see the *IBM Ultrium Tape Drive SCSI Reference*.

WORM (Write once, read many) cartridges

Certain records retention and data security applications require a write once, read many (WORM) method for storing data on tape.

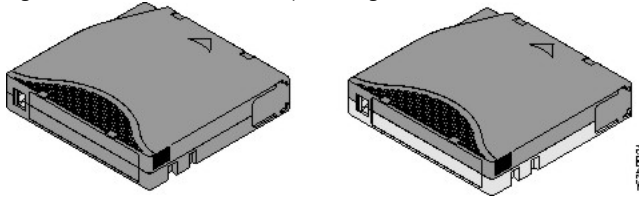
The LTO Ultrium 4 and later drives enable WORM support when a WORM tape cartridge is loaded into the drive.

- [WORM media](#)
- [Data security on WORM media](#)
- [WORM media errors](#)
- [Requirements for WORM capability](#)

WORM media

The standard read/write media are incompatible with the WORM feature so a specially formatted WORM tape cartridge is required, see [Figure 1](#). Each WORM cartridge has a unique, worldwide cartridge identifier (WWCID), which consists of the unique CM chip serial number and the unique tape media serial number. For more information about how to choose and purchase the appropriate WORM tape cartridges for your tape drive, see [Ordering media supplies](#).

Figure 1. Ultrium data and WORM tape cartridges



Data security on WORM media

Certain built-in security measures help ensure that the data that is written on a WORM cartridge does not become compromised, for example:

- The format of an Ultrium WORM tape cartridge is unlike standard read/write media. This unique format prevents a drive that lacks WORM-capable firmware from writing on a WORM tape cartridge. For LTO 9, native data capacity is 18 TB and compressed data capacity is 45 TB.
- When the drive senses a WORM cartridge, the firmware prohibits user data from being changed or altered. The firmware tracks the last point on the tape that can be appended.

WORM media errors

The following conditions cause WORM media errors to occur:

- Information in the servo manufacturer's word (SMW) on the tape must match information from the cartridge memory (CM) module in the cartridge. If it does not match, a media error code 7 posts on the drive's single-character display (SCD).
- Inserting a WORM tape cartridge into a drive that is not compatible with WORM causes the cartridge to be treated as an unsupported medium. The drive reports a media error code 7. Upgrading the drive firmware to the correct code level resolves the problem.

Requirements for WORM capability

To use the WORM capability of your LTO Ultrium drive, you must use a compatible WORM tape cartridge. See [Table 1](#) for cartridge and VOLSER compatibility and [Ordering media supplies](#) for media supplies.

Cleaning cartridge

An IBM® Ultrium Universal Cleaning Cartridge is required to clean the tape drive.

The drive itself determines when it must be cleaned and notifies the library. When notified, the library indicates that the drive needs cleaning by turning ON the "Clean Drive" LED on the front panel of the library and posting a message on the library display.

A tape drive within a library requires the use of a library menu function to either automatically or manually clean the tape drive. See [Methods of cleaning the tape drive](#).

Important: The drive must be cleaned only when it is requested by the drive.

The IBM Ultrium Universal Cleaning Cartridge is valid for 50 uses. The cartridge's LTO-CM (Cartridge Memory) chip tracks the number of times that the cartridge is used.

Note: The drive automatically ejects an expired cleaning cartridge.

Cartridge memory chip (LTO-CM)

All generations of the IBM® LTO Ultrium data cartridges include a Linear Tape-Open Cartridge Memory (LTO-CM) chip.

This chip (1 in [Figure 1](#)) contains information about the cartridge and the tape (such as the name of the manufacturer that created the tape), and statistical information about the cartridge's use. The LTO-CM enhances the efficiency of the cartridge. For example, the LTO-CM stores the end-of-data location which, when the next time this cartridge is inserted and the **Write** command is issued, enables the drive to quickly locate the recording area and begin recording. The LTO-CM also aids in determining the reliability of the cartridge by storing data about its age, how many times it was loaded, and how many errors it accumulated. Whenever a tape cartridge is unloaded, the tape drive writes any pertinent information to the cartridge memory.

The storage capacity of the LTO Generations 9 LTO-CM is 32640 bytes. This capacity is double that of Generations 8, 7, and 6 (16320 bytes), and four times the capacity of Generations 5 and 4 LTO-CM, which is 8160 bytes. LTO Generations 3, 2, and 1 have an LTO-CM capacity of 4096 bytes.

Bar code labels

Bar code label requirements for LTO drives.

A bar code label contains:

- A volume serial number (VOLSER) that is human-readable.
- A bar code that the library can read.

Note: LTO drives do not require cartridges to have bar code labels. Specific library types or models might require cartridges to have bar code labels.

Table 1. Bar code label requirements for Ultrium tape drives and libraries

Ultrium Tape Drive/Library	Bar Code Label Requirements
3572	Required
3573	Required
3576	Required
3580	Not required
3581	Required with optional Bar Code Reader
3582	Required
3583	Required
3584	Required

When read by a library's bar code reader, the bar code identifies the cartridge's VOLSER to the library. The bar code also tells the library whether the cartridge is a data cartridge or cleaning cartridge. In addition, the bar code includes the two-character media-type identifier or M8, or **Lx**, where **x** equals 1, 2, 3, 4, 5, 6, 7, 8, or 9. The letter **x** identifies the cartridge as an LTO cartridge and the number represents the generation of cartridge for that cartridge type. [Figure 1](#) shows a sample bar code label for the LTO Ultrium tape cartridge.

Tape cartridges can be ordered with the labels included or with custom labels. To order tape cartridges and bar code labels, see [Ordering media supplies](#). The bar code for usage in IBM® tape libraries must meet predefined specifications. They include (but are not limited to):

- Eight uppercase alphanumeric characters, where the last 2 characters must be **L9**, **L8**, **L7**, **L6**, **L5**, **L4**, **L3**, **L2**, or **L1** (LZ, LY, LX, LW, LV, LU, or LT for Ultrium WORM cartridges).
- Label and printing to be non-glossy.
- Nominal narrow line or space width of 0.423 mm (0.017 in.).
- Wide to narrow ratio of 2.75:1.
- Minimum bar length of 11.1 mm (0.44 in.).

Table 2. Cartridges and VOLSERS compatible with the Ultrium Tape Drives

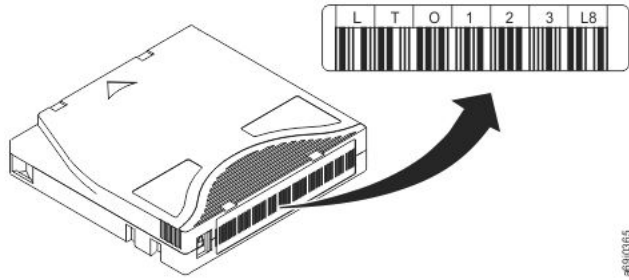
Cartridges	VOLSER
Ultrium 9 Data Cartridge	xxxxxxL9
Ultrium 9 WORM Cartridge	xxxxxxLZ
Ultrium 8 Data Cartridge	xxxxxxL8
Ultrium 8 WORM Cartridge	xxxxxxLY
Ultrium M8 Data Cartridge	xxxxxxM8
Ultrium 7 Data Cartridge	xxxxxxL7
Ultrium 7 WORM Cartridge	xxxxxxLX
Ultrium 6 Data Cartridge	xxxxxxL6
Ultrium 6 WORM Cartridge	xxxxxxLW
Ultrium 5 Data Cartridge	xxxxxxL5
Ultrium 5 WORM Cartridge	xxxxxxLV
Ultrium 4 Data Cartridge	xxxxxxL4
Ultrium 4 WORM Cartridge	xxxxxxLU
Ultrium 3 Data Cartridge	xxxxxxL3
Ultrium 3 WORM Cartridge	xxxxxxLT
Ultrium 2 Data Cartridge	xxxxxxL2
Ultrium 1 Data Cartridge (READ ONLY)	xxxxxxL1
LTO Ultrium Cleaning Cartridge	CLNxxxLx
*An Ultrium 3 Tape Drive must have a minimum firmware level of 54xx for it to be compatible with the WORM cartridge.	

To determine the complete specifications of the bar code and the bar code label, visit the web at <http://www.ibm.com/> and enter "IBM LTO Ultrium Cartridge Label Specification" in the search box, or contact your IBM sales representative.

When a bar code label is attached to a tape cartridge, place the label only in the recessed label area (see 5 in [Figure 1](#)). A label that extends outside of the recessed area can cause loading problems in the drive.

Attention: Do not place any type of mark on the white space at either end of the bar code. A mark in this area might prevent the library from reading the label.

Figure 1. Sample bar code label on the LTO Ultrium 8 Tape Cartridge. The volume serial number (LTO123) and bar code are printed on the label.



- [Guidelines for the use of bar code labels](#)

Guidelines for the use of bar code labels

Apply the following guidelines whenever using bar code labels:

- Use only IBM approved bar code labels on cartridges to be used in an IBM® tape library.
- Do not reuse a label or reapply a used label over an existing label.
- Before you apply a new label, remove the old label by slowly pulling it at a right angle to the cartridge case.
- Use peel-clean labels that do not leave a residue after they are removed. If there is glue residue on the cartridge, remove it by gently rubbing it with your finger. Do not use a sharp object, water, or a chemical to clean the label area.
- Examine the label before it is applied to the cartridge. Do not use the label if it has voids or smears in the printed characters or bar code (a library's inventory operation takes much longer if the bar code label is not readable).
- Remove the label from the label sheet carefully. Do not stretch the label or cause the edges to curl.
- Position the label within the recessed label area (see 5 in [Figure 1](#)).
- With light finger pressure, smooth the label so that no wrinkles or bubbles exist on its surface.
- Verify that the label is smooth and parallel, and has no roll-up or roll-over. The label must be flat to within 0.5 mm (0.02 in.) over the length of the label and have no folds, missing pieces, or smudges.
- Do not place other machine-readable labels on other surfaces of the cartridge. They might interfere with the ability of the drive to load the cartridge.

Write-Protect switch

The position of the write-protect switch on the tape cartridge determines whether you can write to the tape.

If the switch 1 is set to:

- The locked position  (solid red), data cannot be written to the tape.
- The unlocked position (black void), data can be written to the tape.

If possible, use your server's application software to write-protect your cartridges (rather than manually setting the write-protect switch). This application allows the server's software to identify a cartridge that no longer contains current data and is eligible to become a scratch (blank) data cartridge. Do not write-protect scratch (blank) cartridges. The tape drive cannot write new data to them.

If you must manually set the write-protect switch, slide it left or right to the desired position.

Figure 1. Setting the write-protect switch

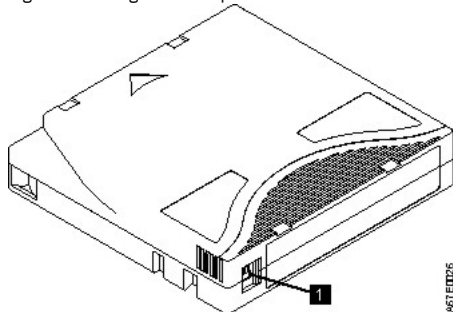


Table 1. Location of the write-protect switch

1	Write-Protect switch

Cartridge care and handling

Incorrect handling or an incorrect environment can damage cartridges or their magnetic tape.

Attention: Do not insert a damaged tape cartridge into the drive. A damaged cartridge can interfere with the reliability of a drive and might void the warranties of the drive and the cartridge. Before inserting a tape cartridge, inspect the cartridge case, cartridge door, and write-protect switch for breaks.

Incorrect handling or an incorrect environment can damage cartridges or their magnetic tape. To avoid damage to your tape cartridges and to ensure the continued high reliability of your IBM® LTO Ultrium Tape Drives, use the following guidelines.

- [Provide training](#)
Providing training to the users of your tape drive can prolong the life of your tape cartridges.
- [Ensuring proper packaging](#)
- [Proper acclimation and environmental conditions](#)
- [Completing a thorough inspection](#)
- [Handling the cartridge carefully](#)
- [Examples of cartridge problems](#)

Provide training

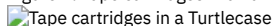
Providing training to the users of your tape drive can prolong the life of your tape cartridges.

- Post procedures that describe proper media handling in places where people gather.
- Ensure that anyone who handles tape has been properly trained in handling and shipping procedures. This includes operators, users, programmers, archival services, and shipping personnel.
- Ensure that any service or contract personnel who perform archiving are properly trained in media-handling procedures.
- Include media-handling procedures as part of any services contract.
- Define and make personnel aware of data recovery procedures.

Ensuring proper packaging

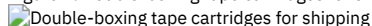
- When a cartridge is shipped, use the original or better packaging.
- Always ship or store a cartridge in a jewel case.
- Use only a recommended shipping container that securely holds the cartridge in its jewel case during transportation. Ultrium Turtlecases (by Perm-A-Store) are tested and found to be satisfactory (see [Figure 1](#)). They are available at <http://www.turtlecase.com>.

Figure 1. Tape cartridges in a Turtlecase



- Never ship a cartridge in a commercial shipping envelope. Always place it in a box or package.
- If you ship the cartridge in a cardboard box or a box of a sturdy material, follow these instructions:
 - Place the cartridge in polyethylene plastic wrap or bags to protect it from dust, moisture, and other contaminants.
 - Pack the cartridge snugly; do not allow it to move around.
 - Double-box the cartridge (place it inside a box, then place that box inside the shipping box) and add padding between the two boxes (see [Figure 2](#)).

Figure 2. Double-boxing tape cartridges for shipping



Proper acclimation and environmental conditions

To ensure the longest possible life for your data cartridges, follow these guidelines.

- Before you use a tape cartridge, acclimate it to the operating environment for 24 hours or the time necessary to prevent condensation in the drive. The time varies, depending on the environmental extremes to which the cartridge was exposed. If the data cartridge was exposed to temperatures outside the specified ranges, stabilize the cartridge at room temperature for the same length of time it was exposed to extreme temperatures or 24 hours, whichever is less.
- Ensure that all surfaces of a cartridge are dry before it is inserted.
- Do not expose data cartridges to direct sunlight or sources of heat, including portable heaters and heating ducts.
- Do not place data cartridges near sources of electromagnetic energy or strong magnetic fields such as computer monitors, electric motors, speakers, or X-ray equipment. Exposure to electromagnetic energy or magnetic fields can destroy data and the embedded servo code that is written on the media by the cartridge manufacturer, which can render the cartridge unusable.
- Use only the data cartridges that are designated for your device.
- Clean the tape drive when the Clean drive LED is illuminated.
- Do not drop a cartridge. Excessive shock can damage the internal contents of the cartridge or the cartridge case itself, making the cartridge unusable.
- Place identification labels only in the designated area on the cartridge.
- Maintain the conditions that are described in [Environmental and shipping specifications for tape cartridges](#).

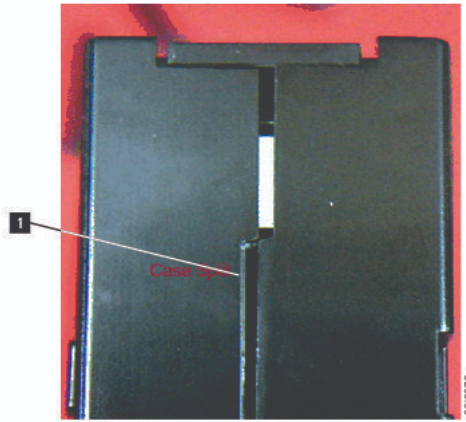
Attention: Do not degauss LTO data cartridges! These data cartridges are pre-recorded with a magnetic servo signal. This signal is required to use the cartridge with the LTO tape drive. Keep magnetically charged objects away from the cartridge.

Completing a thorough inspection

After a cartridge is purchased and before it is used, complete the following steps.

- Inspect the cartridge's packaging to determine potential rough handling.
- When a cartridge is inspected, open only the cartridge door. Do not open any other part of the cartridge case. The upper and lower parts of the case are held together with screws; separating them impairs the usefulness of the cartridge.
- Inspect the cartridge for damage before it is used or stored.
- Inspect the rear of the cartridge (the part that loads first into the tape load compartment) and ensure that no gaps appear in the seam of the cartridge case. See 1 in [Figure 1](#) and 4 in [Figure 2](#). If gaps appear in the seam (see [Figure 1](#)), the leader pin might be dislodged. Go to [Repositioning or reattaching a leader pin](#).

Figure 1. Checking for gaps in the seams of a cartridge



- Check that the leader pin is properly seated. See 2 in [Figure 1](#).
- If you suspect that the cartridge was mishandled but it appears usable, copy any data onto a good cartridge immediately for possible data recovery. Discard the mishandled cartridge.
- Review handling and shipping procedures.

Handling the cartridge carefully

- Do not drop the cartridge. If the cartridge drops, slide the cartridge door back and ensure that the leader pin is properly seated in the pin-retaining spring clips. See 2 in [Figure 1](#). If the leader pin becomes dislodged, go to [Repositioning or reattaching a leader pin](#).
- Do not handle tape that is outside the cartridge. Handling the tape can damage the tape's surface or edges, which might interfere with read or write reliability. Pulling on tape that is outside the cartridge can damage the tape and the brake mechanism in the cartridge.
- Do not stack more than six cartridges.
- Do not degauss a cartridge that you intend to reuse. Degaussing makes the tape unusable.

Examples of cartridge problems

Split cartridge case. See [Figure 1](#).

The cartridge's case is damaged. There is a high possibility of media damage and potential loss.

1. Look for cartridge mishandling.
2. Use the IBM Leader Pin Reattachment Kit (part number 08L9129) to correctly seat the pin (see [Repositioning or reattaching a leader pin](#)). Then, immediately use data recovery procedures to minimize chances of data loss.
3. Review media-handling procedures.

Improper placement of leader pin. See [Figure 1](#).

The leader pin is misaligned.

1. Look for cartridge damage.
2. Use the IBM® Leader Pin Reattachment Kit (part number 08L9129) to correctly seat the pin (see [Repositioning or reattaching a leader pin](#)). Then, immediately use data recovery procedures to minimize chances of data loss.

Repositioning or reattaching a leader pin

Information about repairing your tape cartridges: repositioning or reattaching the leader pins.

Attention: Use a repaired tape cartridge only to recover data and move it to another cartridge. Continued use of a repaired cartridge might void the warranties of the drive and the cartridge.

If the leader pin in your cartridge becomes dislodged from its pin-retaining spring clips or detaches from the tape, you must use the IBM Leader Pin Reattachment Kit (part number 08L9129) to reposition or reattach it. (Do not reattach the pin if you must remove more than 7 meters (23 feet) of leader tape). The sections that follow describe each procedure.

- [Repositioning a leader pin](#)
- [Reattaching a leader pin](#)

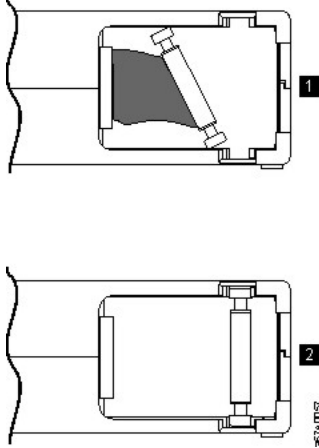
Repositioning a leader pin

A leader pin that is improperly seated inside a cartridge interferes with the operation of the drive. [Figure 1](#) shows a leader pin in the incorrect 1 and correct 2 positions.

To place the leader pin in its proper position, you need the following tools:

- Plastic or blunt-end tweezers
- Cartridge manual rewind tool (from Leader Pin Reattachment Kit, part number 08L9129)

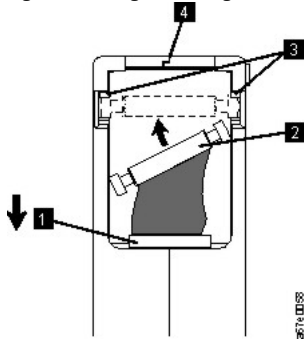
Figure 1. Leader pin in the incorrect and correct positions. The cartridge door is open and the leader pin is visible inside the cartridge.



To reposition the leader pin:

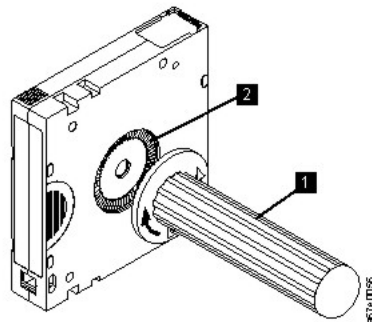
1. Slide open the cartridge door (1 in [Figure 2](#)) and locate the leader pin 2 (you might need to shake the cartridge gently to roll the pin toward the door).
2. With plastic or blunt-end tweezers, grasp the leader pin and position it in the pin-retaining spring clips 3.
3. Press the leader pin gently into the clips until it snaps into place and is firmly seated.
4. Close the cartridge door.

Figure 2. Placing the dislodged leader pin into the correct position. The cartridge door is open to show the leader pin.



5. To rewind the tape, insert the cartridge manual rewind tool (1 in [Figure 3](#)) into the cartridge's hub (2) and turn it clockwise until the tape becomes taut.

Figure 3. Rewinding the tape into the cartridge



6. Remove the rewind tool by pulling it away from the cartridge.
7. If you suspect that the cartridge was mishandled but it appears usable, copy any data onto a good cartridge immediately for possible data recovery. Discard the mishandled cartridge.

Reattaching a leader pin

The first meter of tape in a cartridge is leader tape. Once the leader tape is removed, tape breakage might occur. After the leader pin is reattached, transfer data from the defective tape cartridge. **Do not reuse the defective tape cartridge.**

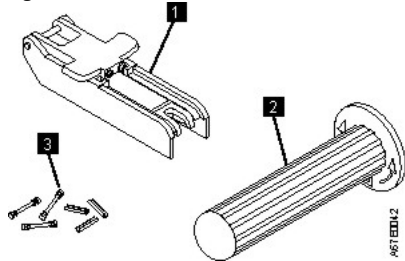
The Leader Pin Reattachment Kit contains three parts.

- **Leader pin attach tool** (see 1 in [Figure 1](#)). A plastic brace that holds the cartridge door open.
- **Cartridge manual rewind tool** (see 2 in [Figure 1](#)). A device that fits into the cartridge's hub and winds the tape into and out of the cartridge.
- **Pin supplies** (see 3 in [Figure 1](#)). Leader pins and C-clips.

Attention:

- Use only the IBM Leader Pin Reattachment Kit to reattach the leader pin to the tape. Other methods of reattaching the pin damages the tape, the drive, or both.
- Use this procedure on your tape cartridge only when the leader pin detaches from the magnetic tape and you must copy the cartridge's data onto another cartridge. Destroy the damaged cartridge after you copy the data. This procedure might affect the performance of the leader pin during threading and unloading operations.
- Touch only the end of the tape. Touching the tape in an area other than the end can damage the tape's surface or edges, which might interfere with read or write reliability.

Figure 1. Leader Pin Reattachment Kit

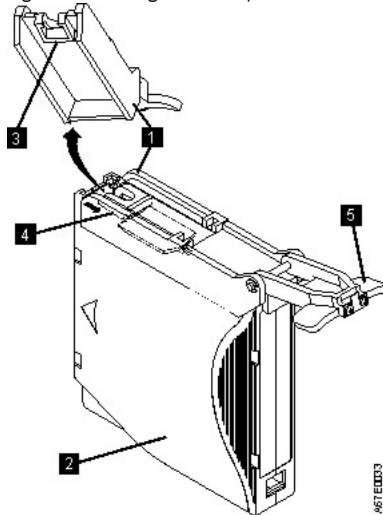


The following procedure describes how to reattach a leader pin.

To reattach a leader pin by using the IBM Leader Pin Reattachment Kit:

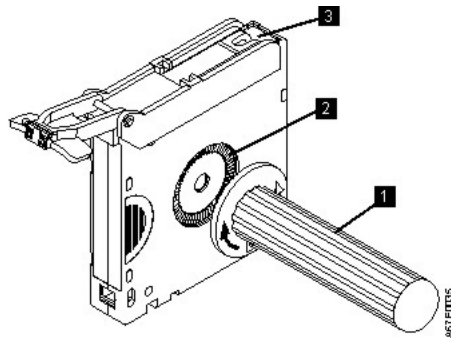
1. Attach the leader pin attach tool (1 in [Figure 2](#)) to the cartridge 2 so that the tool's hook 3 latches into the cartridge's door 4. Pull the tool back to hold the door open, then slide the tool onto the cartridge. Open the tool's pivot arm 5.

Figure 2. Attaching the leader pin attach tool to the cartridge. To hold the cartridge door open, hook the tool into the door and pull the tool back.



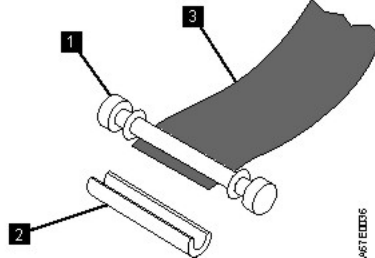
2. To find the end of the tape inside the cartridge, attach the cartridge manual rewind tool (1 in [Figure 3](#)) to the cartridge's hub 2 by fitting the tool's teeth between the teeth of the hub. Turn the tool clockwise until you see the end of the tape inside the cartridge. Then, slowly turn the rewind tool counterclockwise to bring the tape edge toward the cartridge door 3.
3. Continue to turn the rewind tool counterclockwise until approximately 13 cm (5 in.) of tape hangs from the cartridge door. If necessary, grasp the tape and pull gently to unwind it from the cartridge.
4. Remove the rewind tool by pulling it away from the cartridge. Set the tool and the cartridge aside.

Figure 3. Winding the tape out of the cartridge. Turn the cartridge manual rewind tool clockwise to see the end of the tape, then turn it counterclockwise to bring the tape to the cartridge door.



5. On the leader pin (1 in [Figure 4](#)), locate the open side of the C-clip 2. The C-clip is a small black part that secures the tape 3 to the pin.
6. Remove the C-clip from the leader pin by using your fingers to push the clip away from the pin. Set the pin aside and discard the clip.

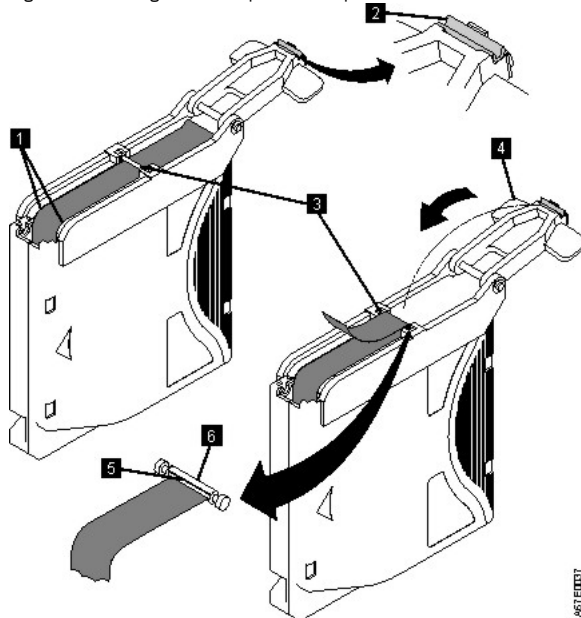
Figure 4. Removing the C-clip from the leader pin. Use your fingers to push the C-clip from the leader pin.



7. Position the tape in the alignment groove of the leader pin attach tool (see 1 in [Figure 5](#)).
8. Place a new C-clip into the retention groove 2 ([Figure 5](#)) on the leader pin attachment tool and make sure that the clip's open side faces up.
9. Place the leader pin (from step 6) into the cavity 3 ([Figure 5](#)) of the leader pin attach tool.
- Attention: To prevent the leader pin from rolling into the cartridge, in the following step use care when the tape is folded over the pin.
10. Fold the tape over the leader pin and hold it with your fingers (see [Figure 5](#)).

Note: Use care to ensure that the tape is centered over the leader pin. Failure to properly center the tape on the pin causes the repaired cartridge to fail. When the tape is properly centered, a 0.25 mm (0.01-in.) gap exists on both sides of the pin.

Figure 5. Attaching the leader pin to the tape



11. Close the pivot arm 4 of the leader pin attach tool by swinging it over the leader pin so that the C-clip snaps onto the pin and the tape.
12. Swing the pivot arm open and trim the excess tape 5 so that it is flush with the reattached leader pin 6.
13. Use your fingers to remove the leader pin from the cavity 3 in the leader pin attach tool.
14. Use the cartridge manual rewind tool to wind the tape back into the cartridge (wind the tape clockwise). Ensure that the leader pin is latched by the pin-retaining spring clips on each end of the leader pin.
15. Remove the rewind tool.
16. Remove the leader pin attach tool by lifting its end up and away from the cartridge.

Environmental and shipping specifications for tape cartridges

Specific storage and shipping environmental conditions apply to tape cartridges.

Before you use a tape cartridge, acclimate it to the operating environment for 24 hours or the time necessary to prevent condensation in the drive. The time varies, depending on the environmental extremes to which the cartridge was exposed.

The best storage container for the cartridges (until they are opened) is the original shipping container. The plastic wrapping prevents dirt from accumulating on the cartridges and partially protects them from humidity changes.

When you ship a cartridge, place it in its jewel case or in a sealed, moisture-proof bag to protect it from moisture, contaminants, and physical damage. Ship the cartridge in a shipping container that has enough packing material to cushion the cartridge and prevent it from moving within the container.

[Table 1](#) gives the environment for storing and shipping LTO Ultrium Tape Cartridges.

Table 1. Environment for storing and shipping the LTO Ultrium Tape Cartridge

Environmental Factor	Allowable Storage	Recommended Storage	Shipping
Temperature	16 - 32°C (61 - 90°F)	16 - 25°C (61 - 77°F)	-23 to 49°C (-9 to 120°F)
Relative humidity (noncondensing)	20 - 80%	20 - 50%	5 - 80%
Maximum wet bulb temperature	26°C (79°F)	26°C (79°F)	26°C (79°F)
Magnetic field	Stray magnetic field at any point on tape not to exceed 50 oersteds (4000 ampere/meter).		

Disposing of tape cartridges

Under the current rules of the US Environmental Protection Agency (EPA), regulation 40CFR261, the LTO Ultrium Tape Cartridge is classified as non-hazardous waste.

As such, it might be disposed of in the same way as normal office trash. These regulations are amended from time to time, and you can review them at the time of disposal.

If your local, state, country (non-US), or regional regulations are more restrictive than EPA 40CFR261, you must review them before you dispose of a cartridge. Contact your account representative for information about the materials that are in the cartridge.

If a tape cartridge must be disposed of in a secure manner, you can erase the data on the cartridge by using a high-energy ac degausser (use a minimum of 4000 oersted peak field over the entire space that the cartridge occupies). The tape must make two passes through the field at 90 degree orientation change for each pass to achieve complete erasure. Some commercial degaussers have one magnetic field regions offset 90 degrees from each other to accomplish complete erasure in one pass for higher throughput. Degaussing makes the cartridge unusable.

If you burn the cartridge and tape, ensure that the incineration complies with all applicable regulations.

Ordering media supplies

[Table 1](#) lists the cartridges and media supplies that you can order for the drive.

Table 1. Media supplies

Supply Item	Methods of Ordering	Feature codes for Radio Frequency Identification (RFID) labels
12 TB Ultrium 8 Data Cartridge 20Pk Bar code labels are preapplied to cartridges.	Specify Machine Type 3589 Model 552 and the VOLSER characters that you want.	Available from media resellers.
12 TB Ultrium 8 Data Cartridge 20Pk Order VOLSER labels separately.	Specify Machine Type 3589 Model 652.	N/A
12 TB Ultrium 8 Data Cartridge 5Pk (black and white labels unattached)	Specify Part Number 01PL340 through an IBM-authorized distributor.	N/A
12 TB Ultrium 8 WORM Data Cartridges	Available from media resellers.	Available from media resellers.
9 TB Ultrium 7 Uninitialized Data Cartridge 20Pk M8 Labeled	Specify Machine Type 3589 Model 452 and the VOLSER characters that you want.	N/A
6 TB Ultrium 7 Data Cartridge 20Pk Bar code labels are preapplied to cartridges.	Specify Machine Type 3589 Model 551 and the VOLSER characters that you want. You can also order through an IBM-authorized distributor.	FC 5511
6 TB Ultrium 7 Data Cartridge 20Pk Order VOLSER labels separately.	Specify Machine Type 3589 Model 651. You can also order through an IBM-authorized distributor.	N/A
6 TB Ultrium 7 Data Cartridge 5Pk No labels.	Specify Part Number 38L7189 through an IBM-authorized distributor.	N/A
6 TB Ultrium 7 WORM Tape Cartridge 20Pk Bar code labels are preapplied to cartridges.	Specify Machine Type 3589 Model 571 and the VOLSER characters that you want. If you would like RFID labels, also specify the feature code provided.	FC 5711
6 TB Ultrium 7 WORM Tape Cartridge 20Pk Order VOLSER labels separately.	Specify Machine Type 3589 Model 671.	N/A

Supply Item	Methods of Ordering	Feature codes for Radio Frequency Identification (RFID) labels
2.5 TB Ultrium 6 Data Cartridge 20Pk Bar code labels are preapplied to cartridges.	Specify Machine Type 3589 Model 550 and the VOLSER characters that you want. If you would like RFID labels, also specify the feature code provided.	FC 5510
2.5 TB Ultrium 6 Data Cartridge 20Pk Order VOLSER labels separately.	Specify Machine Type 3589 Model 650.	N/A
2.5 TB Ultrium 6 Data Cartridge 5Pk (black and white labels unattached)	Specify Part Number 35P1902 through an IBM-authorized distributor	N/A
2.5 TB Ultrium 6 WORM Tape Cartridge 20Pk Bar code labels are preapplied to cartridges.	Specify Machine Type 3589 Model 570 and the VOLSER characters that you want.. If you would like RFID labels, also specify the feature code provided.	FC 5710
2.5 TB Ultrium 6 WORM Tape Cartridge 20Pk Order VOLSER labels separately.	Specify Machine Type 3589 Model 670.	N/A
Leader Pin Reattachment Kit	Order Part Number 08L9129 through an IBM-authorized distributor.	N/A
Manual Rewind Tool	Order Part Number 08L9130 through an IBM-authorized distributor.	N/A

To find the closest IBM-authorized distributor, visit the web at <http://www.ibm.com/storage/media> or call 1-888-IBM-MEDIA.

- [Ordering bar code labels](#)

Ordering bar code labels

LTO drives don't require cartridges to have bar code labels. However, in a library environment it's highly recommended to use bar code labels for performance and easily identifiable reasons. You can order these labels separately from the IBM® data cartridges and cleaning cartridges.

You can order bar code labels directly from the authorized suppliers of bar code labels.

Table 1. Authorized suppliers of custom bar code labels

In the Americas	In Europe and Asia
EDP/Tri-Optic 6800 West 117th Avenue Broomfield, CO 80020 US Telephone: 888-438-8362 or 303-464-3547 Fax: 888-438-8363 or 303-666-2166 http://www.tri-optic.com	
Netc, L.L.C. ¹ 100 Corporate Drive Trumbull, CT 06611 US Telephone: 203-372-6382 Fax: 203-372-0676 http://www.netclabels.com	Netc Europe ¹ Telephone.: +49-2151-970-900 Fax: +49-2151-970-908 Email: Vertrieb@netclabels.de http://www.netclabels.de
	Netc Asia Pacific Pty. Ltd. ¹ 7 Cordwood Drive Cooroy QLD 4563 Australia Telephone: +61 (0)7 5442 6263 Fax: +61 (0)7 5442 6522 http://www.netclabels.com.au
¹ Netc is the only authorized supplier of radio frequency identification (RFID) labels. Orders for RFID labels must be placed through the U.S. office. Orders are shipped worldwide.	

Replacement parts

TS4300 tape library has Tier 1 and Tier 2 CRUs (customer replaceable units). These CRUs are parts of the library that must be added, removed, and replaced by the customer. Tier 1 CRUs do not require tools for installation while Tier 2 CRUs need tools for installation.

To order parts, contact IBM® Service. See [Contacting IBM technical support](#).

Table 1. Replacement parts

Part Number	Description
00VJ943	3555 Base Chassis, 32 slot - Includes operator panel, accessor, spooling mechanism, 1 power supply
02JE720 ¹	3555 Base Chassis, 40 slot - Includes operator panel, accessor, spooling mechanism, 1 power supply
00VJ944	3555 Expansion Chassis

Part Number	Description
00VJ955	3555 Accessor, 32 slot - Includes Spooling Mechanism
02JE722 ²	3555 Accessor, 40 slot - Includes Spooling Mechanism
00VJ956	3555 Spooling Mechanism
02XV763	3555 Base Controller
00VJ962	3555 Expansion Controller
00GH772	3555 Expansion Cable
00GH763	3555 LTO6 FH FC Drive Assembly
00GH761	3555 LTO6 HH FC Drive Assembly
00GH759	3555 LTO6 HH SAS Drive Assembly
00GH768	3555 LTO7 FH FC Drive Assembly
00GH766	3555 LTO7 HH FC Drive Assembly
00GH764	3555 LTO7 HH SAS Drive Assembly
00GH814	3555 LTO8 FH FC Drive Assembly
00GH812	3555 LTO8 HH FC Drive Assembly
00GH810	3555 LTO8 HH SAS Drive Assembly
00GH724	3555 Left Magazine Asm
00GH726	3555 Right Magazine Asm
00VJ940	3555 Power supply
12R9314	Fibre Channel Wrap Tool
95P6566	SAS Wrap Tool
00VJ139	2 M Mini-SAS/Mini-SAS 1x Cable
15R8848	Fiber Cable HBW LC/LC 25 M
41V2120	Fiber Cable HBW LC/LC 10 M
46C2900	4 M AE1 MINI-SAS TO MINI-SAS
46C2902	3 M Y MINI-SAS HD TO MINI-SAS
46X9904	SAS 1X to 4X INTERPOSER
39M5068	Power Cable 10 AMP / 250 VAC - 2.8M
39M5081	Power Cable 10 AMP / 125 VAC - 2.8M
39M5095	Power Cable 10 AMP / 250 VAC - 2.8M
39M5102	Power Cable 10 AMP / 250 VAC - 2.8M
39M5123	Power Cable 10 AMP / 250 VAC - 2.8M
39M5130	Power Cable 10 AMP / 250 VAC - 2.8M
39M5144	Power Cable 10 AMP / 250 VAC - 2.8M
39M5151	Power Cable 10 AMP / 250 VAC - 2.8M
39M5158	Power Cable 10 AMP / 250 VAC - 2.8M
39M5165	Power Cable 10 AMP / 250 VAC - 2.8M
39M5172	Power Cable 10 AMP / 250 VAC - 2.8M
39M5199	Power Cable 10 AMP / 250 VAC - 2.8M
39M5206	Power Cable 10 AMP / 250 VAC - 2.8M
39M5219	Power Cable 10 AMP / 250 VAC - 2.8M
39M5226	Power Cable 10 AMP / 250 VAC - 2.8M
39M5240	Power Cable 10 AMP / 250 VAC - 2.8M
39M5247	Power Cable 10 AMP / 250 VAC - 2.8M
39M5254	Power Cable 10 AMP / 250 VAC - 2.8M
39M5378	Power Cable 10 AMP / 250 VAC - 4.3M
^{1, 2} These parts can only be used in libraries having serial number 7800K0K or greater. Ensure that library firmware is at 1.2.1.0-A00 or later to support newer library serial numbers.	

Manual cartridge removal procedure

Follow this procedure if a cartridge must be manually removed and repaired.

Attention:

- It is recommended that the drive and tape be returned to IBM for removal and recovery.
- If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make no further attempts to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.
- These procedures must be completed only by a trained IBM® service provider. SSRs claim their time against service code 33 ECA 013 when they complete this procedure.
- Inform the customer that the following procedure has high risk of damaging the drive and high risk of not being able to recover the data.
- A drive brick transfer to another drive sled is NOT POSSIBLE with this library.

- [Recommended tools](#)
- [Before you begin](#)
- [Beginning procedure](#)
- [Full height drive: Tape spooled off supply reel](#)
- [Half height drive: Tape spooled off supply reel](#)

- [Full height drive: Tape pulled from or broken near leader pin](#)
- [Half height drive: Tape pulled from or broken near leader pin](#)
- [Full height drive: Tape broken in mid-tape](#)
- [Half height drive: Tape broken in mid-tape](#)
- [Full height drive: Tape tangled along tape path](#)
- [Half height drive: Tape tangled along tape path](#)
- [Full height drive: No apparent failure or damage to tape](#)
- [Half height drive: No apparent failure or damage to tape](#)
- [Ending procedure](#)

Recommended tools

- #1 Phillips screwdriver
- ESD Kit
- Flashlight (optional)
- #1 Flathead screwdriver (optional)

Before you begin

When a tape drive error is reported by the library, the drive produces a memory dump and saves it in Random Access Memory (RAM). If the library or drive is powered OFF, this information is lost. To preserve this information for analysis by IBM Technical Support, the drive memory dump must be copied to the drive's flash memory. Complete the following steps to write a drive memory dump to flash memory.

1. Log in to the Management GUI.
2. Attempt to remove the cartridge with the device power ON and with library manager, a host application, or the Unload button. Press and hold Unload for 12 seconds. This action causes the drive to eject the cartridge when it completes the midtape recovery
3. Ensure that the operator issued the appropriate application commands to complete a rewind and unload of the cartridge. This procedure is to ensure that the stuck cartridge is not because of a hang condition in the application
4. Attempt to remove the cartridge by power-cycling the drive. Look for the drive to attempt a midtape recovery.
Note: It can take 5 minutes to 1 hour (depending on cartridge type (LTO 5, and so on) and how much of the tape is spooled out of the cartridge) for the cartridge to rewind and unload.
5. If the cartridge unloads, inform the operator that the cartridge is unloaded. If the cartridge does not unload, repeat steps 2 and 3 before this procedure is continued.
Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

Beginning procedure

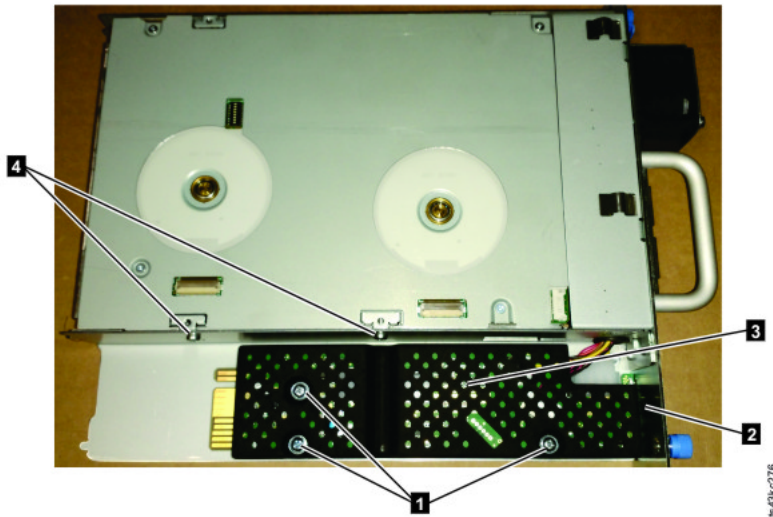
Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

- [Removing the drive brick from the sled](#)
- [Removing the drive cover](#)

Removing the drive brick from the sled

1. Remove the tape drive sled from the library.
 2. Place the sled on a clean, sturdy work surface.
- It is not necessary to remove the plastic cover or connection card.

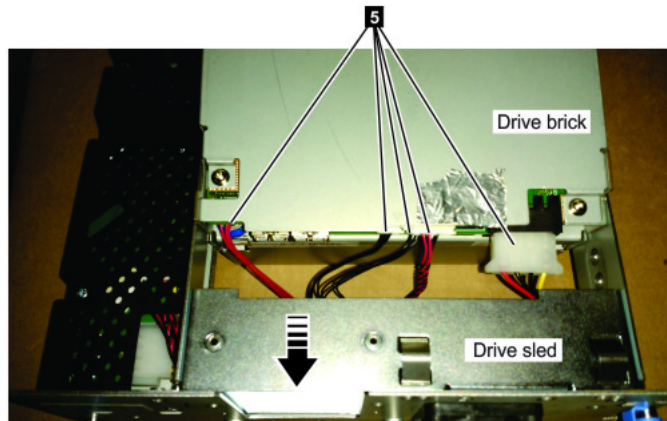
Figure 1. Connection screws and plastic cover



1	Connection card screws
2	Plastic cover
3	Connection card (under the plastic cover)
4	Screws that hold the drive brick to the sled (2 on each side).

3. Remove the drive brick from the sled by completing these steps:
 - a. Remove the four screws (two on each side of the sled 4) that secure the sled to the drive brick.
 - b. Pull the drive brick out of the front of the sled far enough to unplug the cables (5). Some of the cables and connectors are small and can be delicate, so be careful when you are unplugging them.

Figure 2. The drive brick, showing the cables to be unplugged.



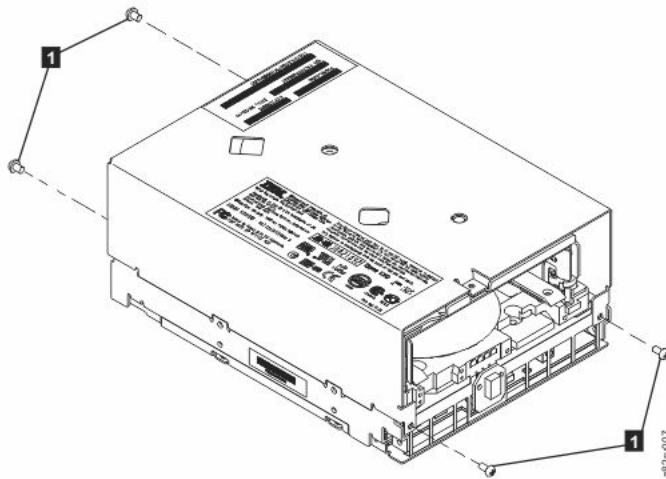
Note: Make sure to note where each cable is connected, so they can be connected correctly later.

- c. Remove the drive brick from the sled.

Removing the drive cover

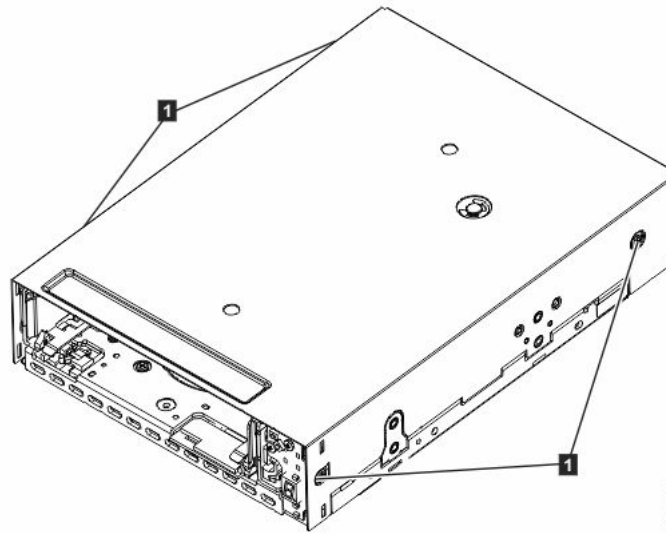
1. Ground yourself to the drive by using an ESD Kit.
2. Remove the cover of the full height drive by completing these steps:
 - a. Remove the four cover-mounting screws and washers 1 .
 - b. Remove the cover by lifting it up.

Figure 1. Removing the cover from the full height drive



3. Remove the cover of the half height drive by completing these steps:
- Remove the four cover-mounting screws (1). Two screws are on each side of the drive.
 - Remove the cover by lifting it up.

Figure 2. Removing the cover from the half height drive



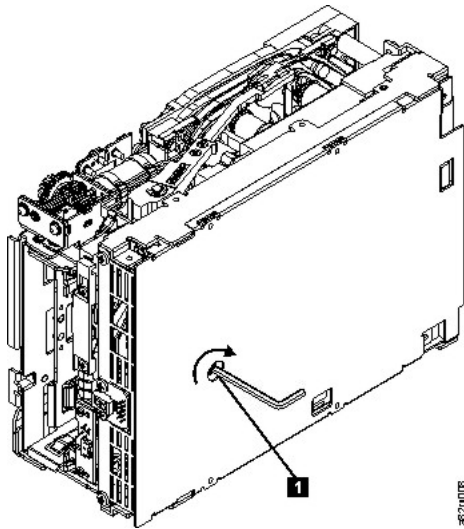
4. Inspect the drive to decide which of the following conditions most closely matches the symptom on the drive:
- Tape spooled off the supply reel - All the tape appears to be on the take up reel and no tape is on the supply reel (inside the cartridge). Test the drive after the procedure is completed.
 - Tape pulled from leader pin (or broken at the front end) - All the tape appears to be on the supply reel (inside the cartridge) and little or no tape appears to be on the take up reel. The leader block is positioned in the take up reel. Return the drive after the procedure is completed.
 - Tape broken in mid-tape - Tape appears to be on both the supply reel (inside the cartridge) and take up reel. Test the drive after the procedure is completed.
 - Tape tangled along tape path - Tape appears to be tangled and damaged but intact. Return the drive after the procedure is completed.
 - OR --
 - No damage to tape (or no apparent failure) - There appears to be no damage or slack to the tape. Return the drive after the procedure is completed.

Full height drive: Tape spooled off supply reel

Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make no further attempts to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

- With the front of the drive facing you, pull an arm's length of tape out of the take up reel from the left side of the drive.
- From the take up reel, thread tape around the rear of the tape path and over the head and rollers on the left side of the drive.
- Set the drive on its left side with the head and tape path facing up.
- Moisten a cotton swab with water and wet approximately 13 mm (0.5 in.) of the tape end and feed it onto the supply reel (inside the cartridge).
- From the bottom of the drive, insert a 2.5 mm offset hex wrench through the bottom cover access hole and into the reel motor axle.

Figure 1. The hex wrench rewinds tape into cartridge



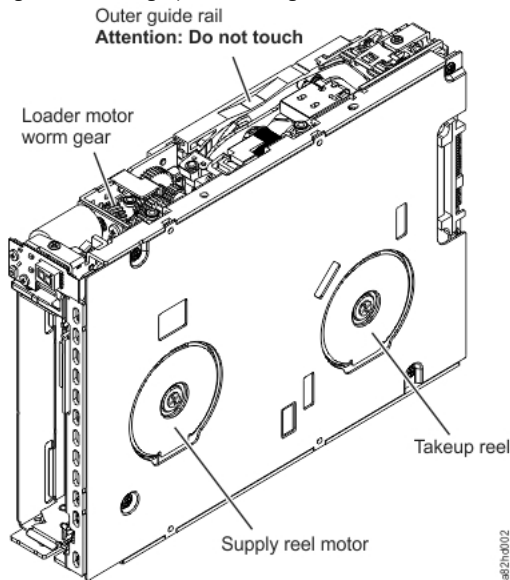
6. Turn the supply reel clockwise, allowing the moistened tape to adhere to the hub as it winds around the supply reel (inside the cartridge).
7. Continue spooling into the cartridge until the tape is taut and remains within the flanges of the tape guiding rollers. Ensure that you do not stretch the tape.
8. Go to [Ending procedure](#).

Half height drive: Tape spooled off supply reel

Attention: **DO NOT TOUCH THE OUTER GUIDE RAIL (2). THIS RAIL IS DELICATE AND EASILY DAMAGED.**
 Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

1. From the takeup reel, pull an arm's length of tape around the rear of the tape path and over the head and rollers on the left side of the drive.
2. Set the drive on its left side with the head and tape path facing up.
3. Ensure that the tape is not twisted. Untwist the tape if required.
4. Moisten a cotton swab with water and wet approximately 13 mm (0.5 in.) of the tape end and feed it onto the supply reel (inside the cartridge).
5. Turn the supply reel (4) clockwise, allowing the moistened tape to adhere to the hub as it winds around the supply reel (inside the cartridge).

Figure 1. Rewinding tape into cartridge



1	Loader motor worm gear	3	Takeup reel motor
2	Outer guide rail (WARNING: Do Not Touch)	4	Supply reel motor

6. Continue spooling into the cartridge until the tape is taut and remains within the flanges of the tape guiding rollers. Turn the supply reel (4) 10 more turns. Ensure that you do not stretch the tape.
7. Go to [Ending procedure](#).

Full height drive: Tape pulled from or broken near leader pin

Figure 1. Drive with cover removed to reveal gear train.

 Drive with cover removed to reveal gear train.

1	Loader motor worm gear	6	Threader mechanism gear
2	Cartridge loader tray guide bearing	7	Lever
3	Rotator stub	8	Loader mechanism gear
4	Threader motor worm gear		
5	Threader intermediate gear		

- From the left side of the drive, pull out tape from the take up reel.
Note: If there is more than approximately 0.6 m (2 ft.) of tape on the take up reel, go to [Full height drive: Tape broken in mid-tape](#).
- If there is less than approximately 0.6 m (2 ft.) of tape on the take up reel, cut off the excess tape as close to the leader pin, as possible.
- Locate the threader motor worm gear (4) the rear of the drive. Use your finger to rotate the threader motor worm gear and slowly rotate the threader mechanism gear (6) clockwise. This action rotates the threader motor worm gear (4) clockwise, drawing the tape leader block assembly (LBA) into the cartridge.
- As the LBA is secured in the cartridge, you hear the LBA retention spring clips click into place. If you do not hear the click, continue rolling until the threader motor worm gear (4) stops. The LBA is in the correct position.
Note: Be sure to keep tension on the tape as the LBA is drawn into the cartridge by using a hex wrench as shown in [Figure 1](#).
- Notice the following mechanisms:
 - Loader mechanism gear (8) nearest the front of the drive that actuates the cartridge loader mechanism
 - Position of the rotator stub (3).
 - Front loader motor worm gear (1). Rotating this gear allows the loader mechanism gear (8) to turn.
- Rotate the loader motor worm gear (1) to turn the loader mechanism gear (6) counterclockwise. Continue turning until the rotator stub (3) loses contact with the lever (7). This action releases the LBA leader pin.
- Rotate the threader motor worm gear (4) to turn the threader mechanism gear (6) counterclockwise. This action moves the LBA out of the cartridge and past the read/write head. Stop this rotation when the LBA is near the tape guide roller nearest the rear of the drive (1).

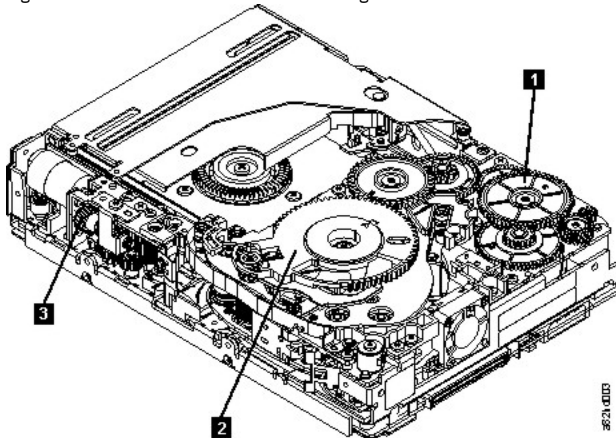
Figure 2. Leader Block Assembly (LBA)

 Leader Block Assembly (LBA)

- Continue rotating the loader motor worm gear (1) until the rotate stub (3) is positioned as shown. Notice that the rotator stub (3) is nearly aligned with the cartridge loader tray guide bearing (2).
- Remove the cartridge from the cartridge loader tray.
- Go to [Ending procedure](#).

Half height drive: Tape pulled from or broken near leader pin

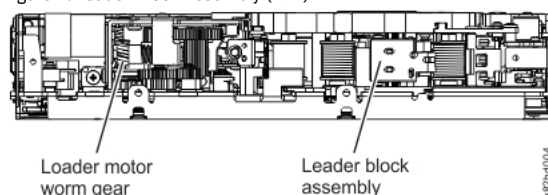
Figure 1. Drive with cover removed to reveal gear train.



1	Threader intermediate gear	2	Threader mechanism gear	3	Loader motor worm gear

- Pull out tape from the take up reel.
Note: If there is more than approximately 0.6 m (2 ft.) of tape on the take up reel, go to [Half height drive: Tape broken in mid-tape](#).
- If there is less than approximately 0.6 m (2 ft.) of tape on the take up reel, cut off the excess tape as close to the leader pin, as possible.
- Reattach the leader pin to the remaining tape.
- Locate the threader intermediate gear (1) near the rear of the drive. You can use your finger to rotate the threader intermediate gear (1) and slowly rotate the threader mechanism gear (2) clockwise. This action draws the tape leader block assembly (LBA) into the cartridge.
- As the leader pin is secured in the cartridge, you hear the leader pin retention spring clips click into place. If you do not hear the click, continue rolling until the threader intermediate gear (1) stops. The LBA is in the correct position.
- Rotate the loader motor worm gear (3) clockwise as viewed from the front of the drive until it stops. This action releases the LBA leader pin.
- Rotate the threader intermediate gear (1) counterclockwise until the leader block is in front of the read/write head. This action moves the LBA out of the cartridge.

Figure 2. Leader Block Assembly (LBA)



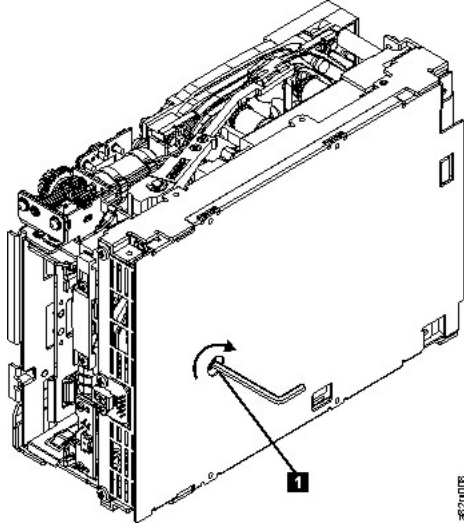
1	Loader motor worm gear	2	Leader block assembly (LBA)

8. Rotate the loader motor worm gear (3) counterclockwise as viewed from the front of the drive until it stops.
9. Remove the cartridge from the cartridge loader tray.
10. Go to [Ending procedure](#).

Full height drive: Tape broken in mid-tape

1. With the front of the drive facing you, pull an arm's length of tape out of the take up reel from the left side of the drive.
Note: If there is less than approximately 5 cm (2 in.) of tape on the take up reel, go to [Full height drive: Tape pulled from or broken near leader pin](#).
2. From the supply reel inside the cartridge, pull approximately 0.3 m (1 ft.) of tape.
3. From the take up reel, thread tape around the rear of the tape path and over the head rollers on the left side of the drive.
4. Moisten a cotton swab with water, and wet approximately 13 mm (0.5 in.) of the tape end. Overlap the tape ends, loosely mending them together.
5. Set the drive on its left side with the head and tape path facing up.
6. From the bottom of the drive, locate the access hole (1 in [Figure 1](#)) in the bottom cover. Insert a 2.5 mm offset hex wrench through the bottom cover access hole and into the reel motor axle. Begin spooling tape back into the cartridge by turning the hex wrench clockwise.

Figure 1. The hex wrench rewinds tape into cartridge

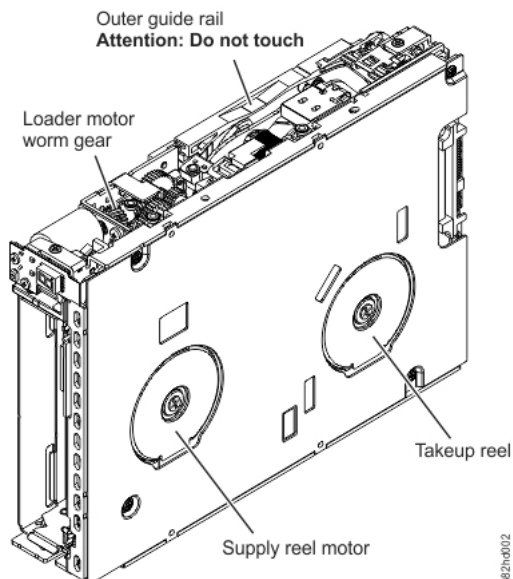


7. Turn the supply reel clockwise, carefully guiding the mended portion of the tape to wind around the hub of the supply reel that is located inside the cartridge. Continue spooling into the cartridge until the tape is taut. The tape must remain within the flanges of the tape guiding rollers. Ensure that you do not stretch the tape.
8. Go to [Ending procedure](#).

Half height drive: Tape broken in mid-tape

1. With the front of the drive facing you, pull an arm's length of tape out of the take up reel. From the take up reel, thread tape around the rear of the tape path and over the head rollers on the left side of the drive.
Note: If there is less than approximately 5 cm (2 in.) of tape on the take up reel, go to [Half height drive: Tape pulled from or broken near leader pin](#).
2. From the supply reel inside the cartridge, pull approximately 0.3 m (1 ft.) of tape.
3. Ensure that the tape is not twisted. Untwist the tape if required.
4. Moisten a cotton swab with water, and wet approximately 13 mm (0.5 in.) of the tape end. Overlap the tape ends, loosely mending them together.
5. Set the drive on its left side with the head and tape path facing up.
6. Turn the supply reel (4) clockwise, carefully guiding the mended portion of the tape to wind around the hub of the supply reel that is located inside the cartridge. Continue spooling into the cartridge until the tape is taut. The tape must remain within the flanges of the tape guiding rollers. Turn the supply reel (4) 10 more turns. Ensure that you do not stretch the tape.

Figure 1. Rewinding tape into cartridge



1	Loader motor worm gear	3	Takeup reel motor
2	Outer guide rail (WARNING: Do Not Touch)	4	Supply reel motor

7. Go to [Ending procedure](#).

Full height drive: Tape tangled along tape path

Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

1. Carefully pull out excess tape and untangle.

Note: If you find the tape to be broken, go to one of the following appropriate procedures:

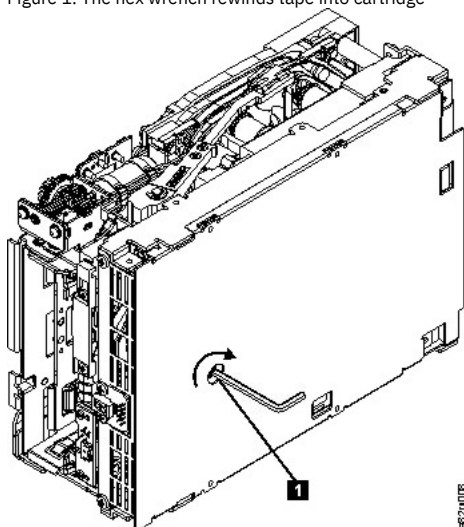
- [Full height drive: Tape spooled off supply reel](#)
- [Full height drive: Tape pulled from or broken near leader pin](#)

--OR--

[Full height drive: Tape broken in mid-tape](#)

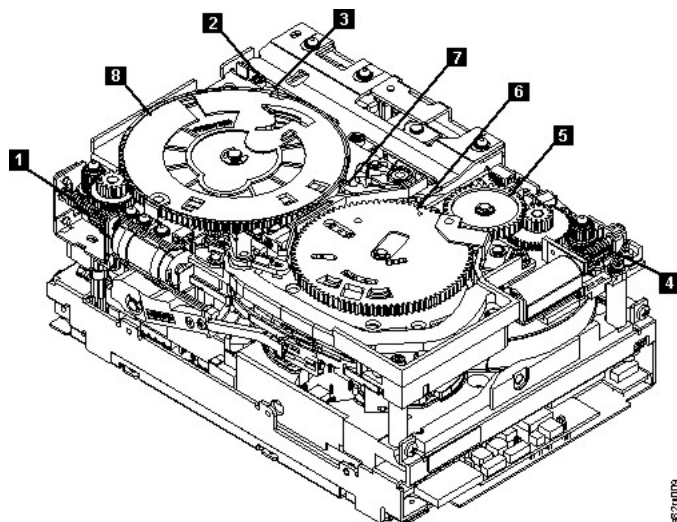
2. Set the drive on its left side with the head and tape path facing up.

Figure 1. The hex wrench rewinds tape into cartridge



3. From the bottom of the drive, locate the access hole (1 in [Figure 1](#)).
4. Insert a 2.5 mm offset hex wrench through the bottom cover access hole and into the reel motor axle. Begin spooling the tape back into the cartridge by turning the hex wrench clockwise.
5. Continue spooling into the cartridge until the tape is taut and remains within the flanges of the tape guiding rollers. Ensure that you do not stretch the tape.
6. Locate the threader motor worm gear (4 in [Figure 2](#)) on the rear of the drive. Use your finger to rotate the threader motor worm gear and slowly rotate the threader mechanism gear (6 in [Figure 2](#)) clockwise.
This action rotates the threader motor worm gear (4 in [Figure 2](#)) clockwise, drawing the LBA into the cartridge.

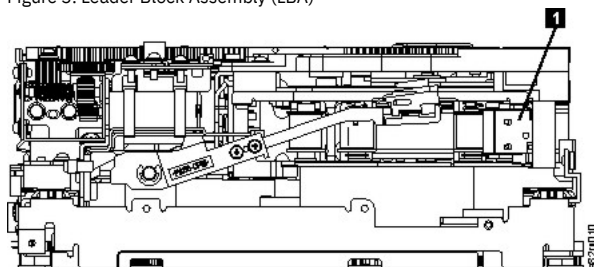
Figure 2. Drive with cover removed to reveal gear train.



1	Loader motor worm gear	6	Threader mechanism gear
2	Cartridge loader tray guide bearing	7	Lever
3	Rotator stub	8	Loader mechanism gear
4	Threader motor worm gear		
5	Threader intermediate gear		

7. As the tape leader block assembly (LBA) is secured in the cartridge, you hear the LBA retention spring clips click into place. If you do not hear the click, continue rolling until the threader motor worm gear (4 in [Figure 2](#)) stops. The LBA is in the correct position.
Note: Be sure to keep tension on the tape as the LBA is drawn into the cartridge by using a hex wrench as shown in [Figure 1](#).
8. Notice these mechanisms:
 - a. Loader mechanism gear (6 in [Figure 2](#)) nearest the front of the drive that actuates the cartridge loader mechanism.
 - b. Position of the rotate stub (3 in [Figure 2](#)).
 - c. Front loader motor worm gear (1 in [Figure 2](#)). Rotating this gear allows the loader mechanism gear (8 in [Figure 2](#)) to turn.
9. Rotate the loader motor worm gear (1 in [Figure 2](#)) to turn the threader mechanism gear (6 in [Figure 2](#)) counterclockwise. Continue turning until the rotator stub (3 in [Figure 2](#)) loses contact with the lever (7 in [Figure 2](#)). This action releases the LBA leader pin.
10. Rotate the threader motor worm gear (4 in [Figure 2](#)) to turn the threader mechanism gear (6 in [Figure 2](#)) counterclockwise. This action moves the LBA out of the cartridge and past the read/write head. Stop this rotation when the LBA is near the tape guide roller nearest the rear of the drive that is shown as 1 [Figure 3](#).

Figure 3. Leader Block Assembly (LBA)



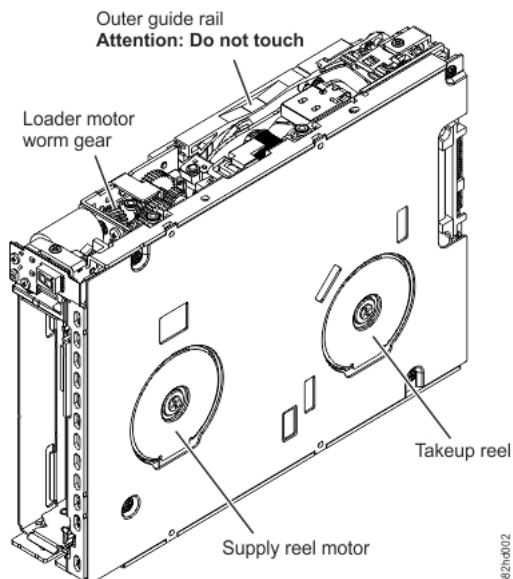
11. Continue rotating the loader motor worm gear (1 in [Figure 2](#)) until the rotator stub (3 in [Figure 2](#)) is positioned as shown. Notice that the rotator stub (3 in [Figure 2](#)) is nearly aligned with the cartridge loader tray guide bearing (2 in [Figure 2](#)).
12. Remove the cartridge from the cartridge loader tray.
13. Go to [Ending procedure](#).

Half height drive: Tape tangled along tape path

Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

1. Carefully pull out excess tape and untangle.
Note: If you find the tape to be broken, go to one of the following appropriate procedures:
 - [Half height drive: Tape spooled off supply reel](#)
 - [Half height drive: Tape pulled from or broken near leader pin](#)
 - OR–
 - [Half height drive: Tape broken in mid-tape](#)
2. Set the drive on its left side with the head and tape path facing up.

Figure 1. Rewinding tape into cartridge



1	Loader motor worm gear	3	Takeup reel motor
2	Outer guide rail (WARNING: Do Not Touch)	4	Supply reel motor

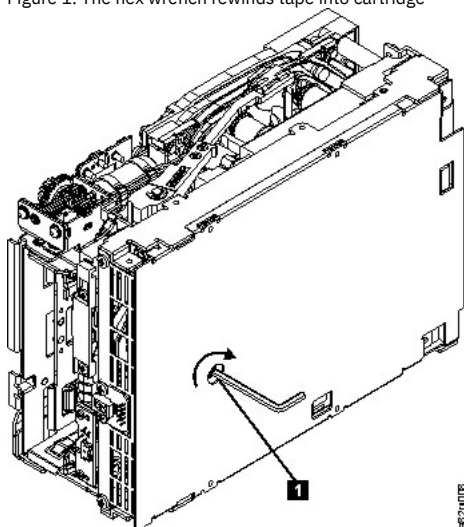
- Turn the supply reel (4) clockwise.
- Continue spooling into the cartridge until the tape is taut and remains within the flanges of the tape guiding rollers. Turn the supply reel (4) 10 turns. Ensure that you do not stretch the tape.
- Go to [Ending procedure](#).

Full height drive: No apparent failure or damage to tape

Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

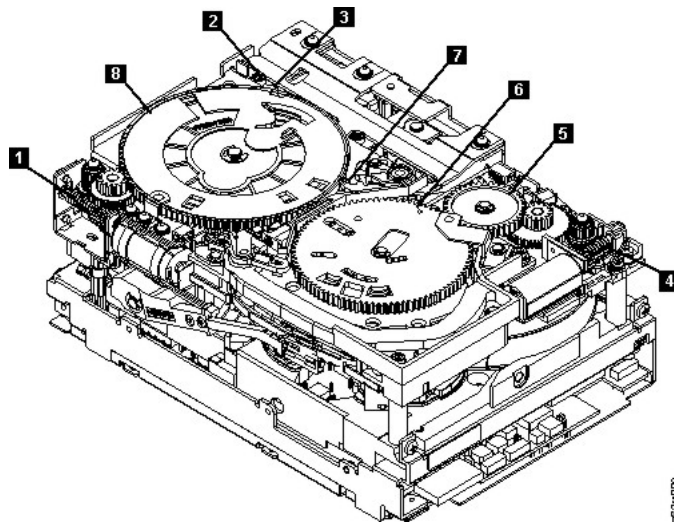
- Set the drive on its left side with the head and tape path facing up.

Figure 1. The hex wrench rewinds tape into cartridge



- From the bottom of the drive, locate the access hole (1 in [Figure 1](#)).
 - Insert a 2.5 mm offset hex wrench through the bottom cover access hole and into the reel motor axle. Begin spooling the tape back into the cartridge by turning the hex wrench clockwise.
 - Continue spooling into the cartridge until the tape is taut and remains within the flanges of the tape guiding rollers. Ensure that you do not stretch the tape.
 - Locate the threader motor worm gear (4 in [Figure 2](#)) on the rear of the drive. Use your finger to rotate the threader motor worm gear and slowly rotate the threader mechanism gear (6 in [Figure 2](#)) clockwise.
- This action rotates the threader motor worm gear (4 in [Figure 2](#)) clockwise, drawing the LBA into the cartridge.

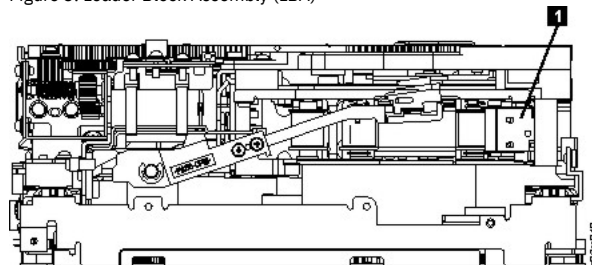
Figure 2. Drive with cover removed to reveal gear train.



1	Loader motor worm gear	6	Threader mechanism gear
2	Cartridge loader tray guide bearing	7	Lever
3	Rotator stub	8	Loader mechanism gear
4	Threader motor worm gear		
5	Threader intermediate gear		

6. As the tape leader block assembly (LBA) is secured in the cartridge, you hear the LBA retention spring clips click into place. If you do not hear the click, continue rolling until the threader motor worm gear (4 in [Figure 2](#)) stops. The LBA is in the correct position.
Note: Be sure to keep tension on the tape as the LBA is drawn into the cartridge by using a hex wrench as shown in [Figure 1](#).
7. Notice these mechanisms:
 - a. Loader mechanism gear (6 in [Figure 2](#)) nearest the front of the drive that actuates the cartridge loader mechanism.
 - b. Position of the rotate stub (3 in [Figure 2](#)).
 - c. Front loader motor worm gear (1 in [Figure 2](#)). Rotating this gear allows the loader mechanism gear (8 in [Figure 2](#)) to turn.
8. Rotate the loader motor worm gear (1 in [Figure 2](#)) to turn the loader mechanism gear (6 in [Figure 2](#)) counterclockwise. Continue turning until the rotator stub (3 in [Figure 2](#)) loses contact with the lever (7 in [Figure 2](#)). This action releases the LBA leader pin.
9. Rotate the threader motor worm gear (4 in [Figure 2](#)) to turn the threader mechanism gear (6 in [Figure 2](#)) counterclockwise. This action moves the LBA out of the cartridge and past the read/write head. Stop this rotation when the LBA is near the tape guide roller nearest the rear of the drive that is shown as 1 [Figure 3](#).

Figure 3. Leader Block Assembly (LBA)



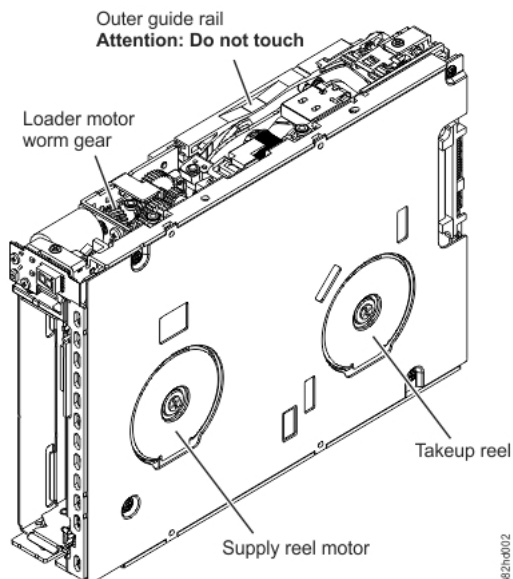
10. Continue rotating the loader motor worm gear (1 in [Figure 2](#)) until the rotator stub (3 in [Figure 2](#)) is positioned as shown. Notice that the rotator stub (3 in [Figure 2](#)) is nearly aligned with the cartridge loader tray guide bearing (2 in [Figure 2](#)).
11. Remove the cartridge from the cartridge loader tray.
12. Go to [Ending procedure](#).

Half height drive: No apparent failure or damage to tape

Note: If the cartridge in the drive is an INPUT tape that contains ACTIVE or 'ONLY COPY' data (there is no backup), eject commands that are issued at the host fail to unload the tape, and power-cycling the drive fails to eject the cartridge, make **no further attempts** to unload this tape. Call Technical Support and open a PMR if one is not already open, to initiate the process of sending the drive with the loaded cartridge in for recovery.

1. Set the drive on its left side with the head and tape path facing up.

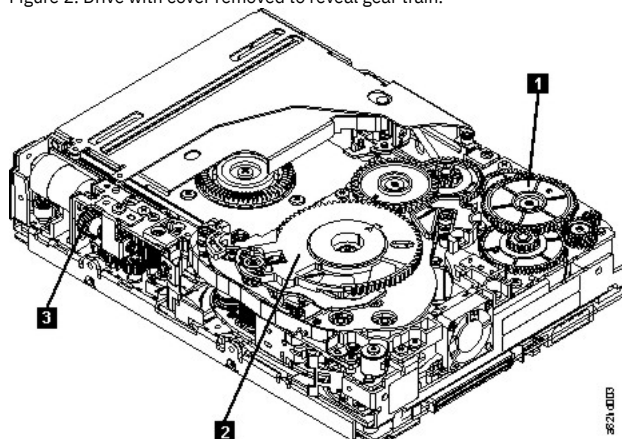
Figure 1. Rewinding tape into cartridge



1	Loader motor worm gear	3	Takeup reel motor
2	Outer guide rail (WARNING: Do Not Touch)	4	Supply reel motor

2. Begin spooling the tape back into the cartridge by turning the supply reel motor (4) clockwise.
3. Continue spooling into the cartridge until the tape is taut and remains within the flanges of the tape guiding rollers. Ensure that you do not stretch the tape. Continue spooling until all tape is removed from the takeup reel (3).
4. Locate the threader intermediate gear (1) near the rear of the drive. You can use your finger to rotate the threader intermediate gear (1) and slowly rotate the threader mechanism gear (2) clockwise. This action draws the tape leader block assembly (LBA) into the cartridge.

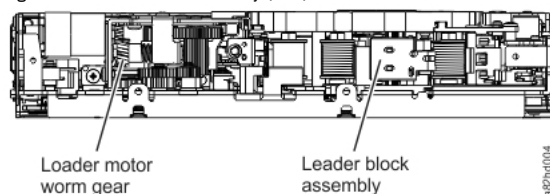
Figure 2. Drive with cover removed to reveal gear train.



1	Threader intermediate gear	2	Threader mechanism gear	3	Loader motor worm gear

5. As the leader pin is secured in the cartridge, you hear the leader pin retention spring clips click into place. If you do not hear the click, continue rolling until the threader intermediate gear (1) stops. The LBA is in the correct position.
Note: Be sure to keep tension on the tape as the LBA is drawn into the cartridge.
6. Rotate the loader intermediate gear (1) clockwise as viewed from the front of the drive until it stops. This action releases the LBA leader pin.
7. Rotate the threader motor worm gear (3) counterclockwise until the leader block is in front of the read/write head. This action moves the LBA out of the cartridge.

Figure 3. Leader Block Assembly (LBA)



1	Loader motor worm gear	2	Leader block assembly (LBA)

8. Rotate the loader motor worm gear (3) counterclockwise as viewed from the front of the drive until it stops.
9. Remove the cartridge from the cartridge loader tray.
10. Go to [Ending procedure](#).

Ending procedure

1. Reassemble the drive brick by reversing the steps in [Removing the drive cover](#).
2. Reassemble the drive sled and reinstall the drive brick by completing these steps:
 - Plug in the RS-422 cable, the power cable (if applicable), and the signal cable.
 - Push the drive brick fully into the sled.
 - Align the two screws holes on each side of the sled with the screw holes on each side of the drive brick. DO NOT TIGHTEN THE SCREWS COMPLETELY.
 - When the drive and sled are properly aligned, fully tighten the screws.
3. Install the tape drive sled in the library.
4. Power ON the library and wait for the library to finish POST (power on self test), inventory, and mid-tape recovery. This step can take up to 1 hour.
5. If the cartridge does not eject from the drive, move the cartridge from the drive to the I/O station, then discard the cartridge.
 - Operator Panel: Operation > Move Cartridge from Drive to Home Slot
 - Management GUI: Drives > Actions > Eject Cartridge from Drive
6. Run Library Verify before normal library operations resume.
7. If necessary, return the failed drive sled to IBM.

Accessibility

Accessibility features help a user with a physical disability, such as restricted mobility or limited vision, to use the HTML version of the customer documentation successfully.

Features

The major accessibility features for the HTML version of the *IBM® TS2900 Tape Autoloader online documentation* are:

- You can use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen. The following screen readers are tested: WebKing and Window-Eyes.
- You can operate all features with the keyboard instead of the mouse.

Navigating by keyboard

You can use keys or key combinations to perform operations and initiate many menu actions that can also be done through mouse actions. You can navigate the HTML version of the *IBM TS2900 Tape Autoloader online documentation* help system from the keyboard. Use the following keyboard combinations:

- To traverse to the next link, button, or topic, press Tab inside a frame (page).
- To move to the previous topic, press ^ or Shift+Tab.
- To scroll all the way up or down, press Home or End.
- To print the current page or active frame, press Ctrl+P.
- To select, press Enter.

Notices

This information was developed for products and services that are offered in the US. This material might be available from IBM in other languages. However, you might be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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 the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

The client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

IBM Privacy Policy

We intend to protect your personal information and to maintain its integrity. IBM® implements reasonable physical, administrative, and technical safeguards to help us protect your personal information from unauthorized access, use, and disclosure. For example, the products only provide IBM data about the asset usage and configuration and do not reflect private use of the asset. When diagnostics are required to be sent to IBM, and a problem is submitted, that data is routed directly to a secured infrastructure. Only individuals with a need to know are given access while working to resolve your problem. When appropriate, we also require that our suppliers protect such information from unauthorized access, use, and disclosure.

Visit the IBM Privacy Policy for additional information on this topic at <https://www.ibm.com/privacy/details/us/en/>.

- **Trademarks**
A list of trademarks for IBM and other companies.
- **Terms and conditions for product documentation**
Permissions for the use of these publications are granted subject to the following terms and conditions.
- **Homologation statement**
This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact an IBM representative or reseller for any questions.
- **Electromagnetic compatibility notices**
- **Safety and environmental notices**
Safety and environmental notices for this product are shown and described.

Trademarks

A list of trademarks for IBM and other companies.

IBM®, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

IBM Privacy Policy

At IBM® we recognize the importance of protecting your personal information and are committed to processing it responsibly and in compliance with applicable data protection laws in all countries in which IBM operates.

Visit the IBM Privacy Policy for additional information on this topic at <https://www.ibm.com/privacy/details/us/en/>.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You cannot distribute, display, or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You can reproduce, distribute, and display these publications solely within your enterprise provided that all proprietary notices are preserved. You cannot make derivative works of these publications, or reproduce, distribute, or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses, or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions that are granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or as determined by IBM, the above instructions are not being properly followed.

You cannot download, export, or reexport this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at <http://www.ibm.com/legal/copytrade.shtml>.

Homologation statement

This product may not be certified in your country for connection by any means whatsoever to interfaces of public telecommunications networks. Further certification may be required by law prior to making any such connection. Contact an IBM representative or reseller for any questions.

Electromagnetic compatibility notices

The following Class A statements apply to IBM® products and their features unless designated as electromagnetic compatibility (EMC) Class B in the feature information.

When attaching a monitor to the equipment, you must use the designated monitor cable and any interference suppression devices that are supplied with the monitor.

- [Canada Notice](#)
- [European Community and Morocco Notice](#)
- [Germany Notice](#)
- [Japan Electronics and Information Technology Industries Association \(JEITA\) Notice](#)
- [Japan Voluntary Control Council for Interference \(VCCI\) Notice](#)
- [Korea Notice](#)
- [People's Republic of China Notice](#)
- [Russia Notice](#)
- [Taiwan Notice](#)
- [United Kingdom Notice](#)
- [United States Federal Communications Commission \(FCC\) Notice](#)

Canada Notice

CAN ICES-3 (A)/NMB-3(A)

European Community and Morocco Notice

This product is in conformity with the protection requirements of Directive 2014/30/EU of the European Parliament and of the Council on the harmonization of the laws of the Member States relating to electromagnetic compatibility. IBM cannot accept responsibility for any failure to satisfy the protection requirements resulting from a non-recommended modification of the product, including the fitting of non-IBM option cards.

This product may cause interference if used in residential areas. Such use must be avoided unless the user takes special measures to reduce electromagnetic emissions to prevent interference to the reception of radio and television broadcasts.

Warning: This equipment is compliant with Class A of CISPR 32. In a residential environment this equipment may cause radio interference.

Germany Notice

Deutschsprachiger EU Hinweis: Hinweis für Geräte der Klasse A EU-Richtlinie zur Elektromagnetischen Verträglichkeit

Dieses Produkt entspricht den Schutzanforderungen der EU-Richtlinie 2014/30/EU zur Angleichung der Rechtsvorschriften über die elektromagnetische Verträglichkeit in den EU-Mitgliedsstaaten und hält die Grenzwerte der EN 55032 Klasse A ein.

Um dieses sicherzustellen, sind die Geräte wie in den Handbüchern beschrieben zu installieren und zu betreiben. Des Weiteren dürfen auch nur von der IBM empfohlene Kabel angeschlossen werden. IBM übernimmt keine Verantwortung für die Einhaltung der Schutzanforderungen, wenn das Produkt ohne Zustimmung von IBM verändert bzw. wenn Erweiterungskomponenten von Fremdherstellern ohne Empfehlung von IBM gesteckt/eingebaut werden.

EN 55032 Klasse A Geräte müssen mit folgendem Warnhinweis versehen werden:

"Warnung: Dieses ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funk-Störungen verursachen; in diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen zu ergreifen und dafür aufzukommen."

Deutschland: Einhaltung des Gesetzes über die elektromagnetische Verträglichkeit von Geräten

Dieses Produkt entspricht dem "Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG)." Dies ist die Umsetzung der EU-Richtlinie 2014/30/EU in der Bundesrepublik Deutschland.

Zulassungsbescheinigung laut dem Deutschen Gesetz über die elektromagnetische Verträglichkeit von Geräten (EMVG) (bzw. der EMC Richtlinie 2014/30/EU) für Geräte der Klasse A

Dieses Gerät ist berechtigt, in Übereinstimmung mit dem Deutschen EMVG das EG-Konformitätszeichen - CE - zu führen.

Verantwortlich für die Einhaltung der EMV-Vorschriften ist der Hersteller:

International Business Machines Corp.
New Orchard Road
Armonk, New York 10504
Tel: 914-499-1900

Der verantwortliche Ansprechpartner des Herstellers in der EU ist:

IBM Deutschland GmbH
Technical Relations Europe, Abteilung M456
IBM-Allee 1, 71139 Ehningen, Germany
Tel: +49 800 225 5426
e-mail: Halloibm@de.ibm.com

Generelle Informationen:

Japan Electronics and Information Technology Industries Association (JEITA) Notice

(一社) 電子情報技術産業協会 高調波電流抑制対策実施
要領に基づく定格入力電力値 : IBM Documentationの各製品
の仕様ページ参照

This statement applies to products less than or equal to 20 A per phase.

高調波電流規格 JIS C 61000-3-2 適合品

This statement applies to products greater than 20 A, single phase.

高調波電流規格 JIS C 61000-3-2 準用品

本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。

- 回路分類 : 6 (単相、P F C回路付)
- 換算係数 : 0

This statement applies to products greater than 20 A per phase, three-phase.

高調波電流規格 JIS C 61000-3-2 準用品

本装置は、「高圧又は特別高圧で受電する需要家の高調波抑制対策ガイドライン」対象機器（高調波発生機器）です。

- 回路分類 : 5 (3相、P F C回路付)
- 換算係数 : 0

Japan Voluntary Control Council for Interference (VCCI) Notice

この装置は、クラスA情報技術装置です。この装置を家庭環境で使用すると電磁妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

VCCI-A

Korea Notice

이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다.

People's Republic of China Notice

警告

此为A级产品，在生活环境中，该产品可能会造成无线电干扰。在这种情况下，可能需要用户对其干扰采取切实可行的措施。

Russia Notice

ВНИМАНИЕ! Настоящее изделие относится к классу A. В жилых помещениях оно может создавать радиопомехи, для снижения которых необходимы дополнительные меры

012501

Taiwan Notice

警告使用者：
此為甲類資訊技術設備，
於居住環境中使用時，可
能會造成射頻擾動，在此
種情況下，使用者會被要
求採取某些適當的對策。

IBM Taiwan Contact Information:

台灣IBM 產品服務聯絡方式：
台灣國際商業機器股份有限公司
台北市松仁路7號3樓
電話：0800-016-888

TC000790

United Kingdom Notice

This product may cause interference if used in residential areas. Such use must be avoided unless the user takes special measures to reduce electromagnetic emissions to prevent interference to the reception of radio and television broadcasts.

United States Federal Communications Commission (FCC) Notice

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. IBM® is not responsible for any radio or television interference caused by using other than recommended cables and connectors, or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) this device might not cause harmful interference, and (2) this device must accept any interference received, including interference that might cause undesired operation.

Responsible Party:
International Business Machines Corporation
New Orchard Road
Armonk, NY 10504
Contact for FCC compliance information only: fccinfo@us.ibm.com

Safety and environmental notices

Safety and environmental notices for this product are shown and described.

- [Safety notices](#)
Observe the safety notices when this product is used. These safety notices contain danger and caution notices. These notices are sometimes accompanied by symbols that represent the severity of the safety condition.
- [Possible safety hazards](#)
- [Laser safety and compliance](#)
- [Performing the safety inspection procedure](#)
- [Rack safety](#)
- [Power cords](#)
For your safety, IBM® provides a power cord with a grounded attachment plug to use with this IBM product. To avoid electrical shock, always use the power cord and plug with a properly grounded outlet.
- [Cautions and regulatory compliance statements for NEBS](#)
This library is NEBS certified. This section includes the cautions and regulatory compliance statements for the Network Equipment-Building System (NEBS) certification from the Telcordia Electromagnetic Compatibility and Electrical Safety - Generic Criteria for Network Telecommunications Equipment (A Module of LSSGR, FR-64; TSGR, FR-440; and NEBSFR, FR-2063) Telcordia Technologies Generic Requirements, GR-1089-CORE, Issue 4, June 2006.

Safety notices

Observe the safety notices when this product is used. These safety notices contain danger and caution notices. These notices are sometimes accompanied by symbols that represent the severity of the safety condition.

Most danger or caution notices contain a reference number (Dxxx or Cxxx). Use the reference number to check the translation in the [IBM Safety Notices](#) (G229-9054) manual.

The sections that follow define each type of safety notice and give examples.

Danger notice

A danger notice calls attention to a situation that is potentially lethal or extremely hazardous to people. A lightning bolt symbol always accompanies a danger notice to represent a dangerous electrical condition. A sample danger notice follows:

	DANGER: An electrical outlet that is not correctly wired could place hazardous voltage on metal parts of the system or the devices that attach to the system. It is the responsibility of the customer to ensure that the outlet is correctly wired and grounded to prevent an electrical shock. (D004)

Caution notice

A caution notice calls attention to a situation that is potentially hazardous to people because of some existing condition, or to a potentially dangerous situation that might develop because of some unsafe practice. A caution notice can be accompanied by one of several symbols:

If the symbol is...	It means...
	A hazardous electrical condition with less severity than electrical danger.
	A hazardous condition that is not represented by other safety symbols.
	This product contains a Class II laser. Do not stare into the beam. (C029) Laser symbols are always accompanied by the classification of the laser as defined by the U. S. Department of Health and Human Services (for example, Class I, Class II).
	A hazardous condition due to mechanical movement in or around the product.
	This part or unit is heavy but has a weight smaller than 18 kg (39.7 lb). Use care when lifting, removing, or installing this part or unit. (C008)
	A hazardous condition due to the unit's susceptibility to electrostatic discharge.

Sample caution notices follow:

CAUTION:

The battery is a lithium ion battery. To avoid possible explosion, do not burn. Exchange only with the IBM-approved part. Recycle or discard the battery as instructed by local regulations. In the United States, IBM® has a process for the collection of this battery. For information, call 1-800-426-4333. Have the IBM part number for the battery unit available when you call. (C007)

CAUTION:

The system contains circuit cards, assemblies, or both that contain lead solder. To avoid the release of lead (Pb) into the environment, do not burn. Discard the circuit card as instructed by local regulations. (C014)

CAUTION:

When the Modular Refrigeration Unit (MRU) is removed, immediately remove any oil residue from the MRU support shelf, floor, and any other area to prevent injuries because of slips or falls. Do not use refrigerant lines or connectors to lift, move, or remove the MRU. Use handholds as instructed by service procedures. (C016)

CAUTION:

Do not connect an IBM control unit directly to a public optical network. The customer must use an extra connectivity device between an IBM control unit optical adapter (that is, fibre, ESCON, FICON®) and an external public network. Use a device such as a patch panel, a router, or a switch. You do not need an extra connectivity device for optical fibre connectivity that does not pass through a public network.

Possible safety hazards

Possible safety hazards to the operation of this product are:

Electrical

An electrically charged frame can cause serious electrical shock.

Mechanical

Hazards (for example, a safety cover missing) are potentially harmful to people.

Chemical

Do not use solvents, cleaners, or other chemicals that are not approved for use on this product.

Before the library is used, repair any of the preceding problems.

Laser safety and compliance

Table 1. Class I Laser Product

	The library might contain a laser assembly that complies with the performance standards set by the US Food and Drug Administration for a Class I laser product. Class I laser products do not emit hazardous laser radiation. The library has the necessary protective housing and scanning safeguards to ensure that laser radiation is inaccessible during operation or is within Class I limits. External safety agencies have reviewed the library and have obtained approvals to the latest standards as they apply.

Performing the safety inspection procedure

Before you service the unit, complete the following safety inspection procedure.

1. Stop all activity between the host and the library's tape drive.
2. Turn off the power to the library by switching the Power button on the rear of the tape library to the Off position.
3. Disconnect the tape drive's SAS cable.
4. Unplug the library's power cord from the electrical outlet and the library's power supply unit.
5. Check the library's power cords for damage, such as a pinched, cut, or frayed cord.
6. Check the tape drive's SAS cable for damage.
7. Check the cover of the library for sharp edges, damage, or alterations that expose its internal parts.
8. Check the cover of the library for proper fit. It should be in place and secure.
9. Check the product label at the rear of the library to make sure that it matches the voltage at your outlet.

Rack safety

The following general safety information must be used for all rack-mounted devices.

DANGER



Observe the following precautions when working on or around your IT rack system.

- Heavy equipment - personal injury or equipment damage might result if mishandled.
- Always lower the leveling pads on the rack cabinet.
- Always install stabilizer brackets on the rack cabinet.
- To avoid hazardous conditions due to uneven mechanical loading, always install the heaviest devices in the bottom of the rack cabinet. Always install servers and optional devices starting from the bottom of the rack cabinet.
- Rack-mounted devices are not to be used as shelves or work spaces. Do not place objects on top of rack-mounted devices. In addition, do not lean on rack mounted devices and do not use them to stabilize your body position (for example, when working from a ladder).



- Each rack cabinet might have more than one power cord.
 - For AC powered racks, be sure to disconnect all power cords in the rack cabinet when directed to disconnect power during servicing.
 - For racks with a DC power distribution panel (PDP), turn off the circuit breaker that controls the power to the system unit(s), or disconnect the customer's DC power source, when directed to disconnect power during servicing.
- Connect all devices installed in a rack cabinet to power devices installed in the same rack cabinet. Do not plug a power cord from a device installed in one rack cabinet into a power device installed in a different rack cabinet.
- An electrical outlet that is not correctly wired could place hazardous voltage on the metal parts of the system or the devices that attach to the system. It is the responsibility of the customer to ensure that the outlet is correctly wired and grounded to prevent an electrical shock. (R001 part 1 of 2)

Caution



- Do not install a unit in a rack where the internal rack ambient temperatures might exceed the manufacturer's recommended ambient temperature for all your rack-mounted devices.
- Do not install a unit in a rack where the air flow is compromised. Ensure that air flow is not blocked or reduced on any side, front, or back of a unit that is used for air flow through the unit.
- Consideration must be given to the connection of the equipment to the supply circuit so that overloading of the circuits does not compromise the supply wiring or overcurrent protection. To provide the correct power connection to a rack, refer to the rating labels on the equipment in the rack to determine the total power requirement of the supply circuit.

- (For sliding drawers) Do not pull out or install any drawer or feature if the rack stabilizer brackets are not attached to the rack. Do not pull out more than one drawer at a time. The rack might become unstable if you pull out more than one drawer at a time.



- (For fixed drawers) This drawer is a fixed drawer and must not be moved for servicing unless specified by the manufacturer. Attempting to move the drawer partially or out of the rack might cause the rack to become unstable or cause the drawer to fall out of the rack. (R001 part 2 of 2)

Caution



Removing components from the upper positions in the rack cabinet improves rack stability during relocation. Follow these general guidelines whenever you relocate a populated rack cabinet within a room or building:

- Reduce the weight of the rack cabinet by removing equipment, starting at the top of the rack cabinet. When possible, restore the rack cabinet to the configuration of the rack cabinet as you received it. If this configuration is not known, you must do the following:
 - Remove all devices in the 32U position (compliance ID RACK-001) or 22U (compliance ID RR001) and above.
 - Ensure that the heaviest devices are installed in the bottom of the rack cabinet.
 - Ensure that there are little-to-no empty U-levels between devices installed in the rack-cabinet below the 32U (compliance ID RACK-001) or 22U (compliance ID RR001) level, unless the received configuration specifically allowed it.
- If the rack cabinet you are relocating is part of a suite of rack cabinets, detach the rack cabinet from the suite.
- If the rack cabinet you are relocating was supplied with removable outriggers, they must be reinstalled before the cabinet is relocated.
- Inspect the route that you plan to take to eliminate potential hazards.
- Verify that the route that you choose can support the weight of the loaded rack cabinet. Refer to the documentation that comes with your rack cabinet for the weight of a loaded rack cabinet.
- Verify that all door openings are at least 760 x 2032 mm (30 x 80 in.).
- Ensure that all devices, shelves, drawers, doors, and cables are secure.
- Ensure that the four leveling pads are raised to their highest position.
- Ensure that no stabilizer bracket is installed on the rack cabinet during movement.
- Do not use a ramp that is inclined at more than 10 degrees.
- When the rack cabinet is in the new location, complete these steps.
 - Lower the four leveling pads.
 - Install stabilizer brackets on the rack cabinet or in an earthquake environment bolt the rack to the floor.
 - If you removed any devices from the rack cabinet, repopulate the rack cabinet from the lowest position to the highest position.
- If a long-distance relocation is required, restore the rack cabinet to the configuration of the rack cabinet as you received it. Pack the rack cabinet in the original packaging material, or equivalent. Also, lower the leveling pads to raise the casters off the pallet and bolt the rack cabinet to the pallet. (R002)
- DANGER
Racks with a total weight of > 227 kg (500 lb.), Use Only Professional Movers! (R003)

Caution



- Rack is not intended to serve as an enclosure and does not provide any degrees of protection required of enclosures.
- It is intended that equipment installed within this rack will have its own enclosure. (R005)
- Tighten the stabilizer brackets until they are flush against the rack. (R006)
- Use safe practices when lifting. (R007)
- Do not place any object on top of a rack-mounted device unless that rack-mounted device is intended for use as a shelf. (R008)
- If the rack is designed to be coupled to another rack only the same model rack should be coupled together with another same model rack. (R009)

Danger



Main Protective Earth (Ground): This symbol is marked on the frame of the rack. The PROTECTIVE EARTHING CONDUCTORS must be terminated at that point. A recognized or certified closed loop connector (ring terminal) must be used and secured to the frame with a lock washer using a bolt or stud. The connector must be properly sized to be suitable for the bolt or stud, the locking washer, the rating for the conducting wire used, and the considered rating of the breaker. The intent is to ensure the frame is electrically bonded to the PROTECTIVE EARTHING CONDUCTORS. The hole that the bolt or stud goes into where the terminal connector and the lock washer contact must be free of any non-conductive material to allow for metal to metal contact. All PROTECTIVE BONDING CONDUCTORS must terminate at this main

protective earthing terminal or at points marked with . (R010)



Always ensure that a load of 95 kg (210 lb) is inside the bottom of the rack (compliance ID RR001), especially before relocating or servicing units with their Center of Gravity (CoG) higher than 22U. (R011)

Power cords

For your safety, IBM® provides a power cord with a grounded attachment plug to use with this IBM product. To avoid electrical shock, always use the power cord and plug with a properly grounded outlet.



For your safety, IBM provides a power cord with a grounded attachment plug to use with this IBM product. To avoid electrical shock, always use the power cord and plug with a properly grounded outlet.

IBM power cords that are used in the United States and Canada are listed by Underwriter's Laboratories (UL) and certified by the Canadian Standards Association (CSA).

For units intended to be operated at 115 volts: Use a UL-listed and CSA-certified cord set consisting of a minimum 18 AWG, Type SVT or SJT, three-conductor cord, a maximum of 15 feet in length and a parallel blade, grounding-type attachment plug rated 15 amperes, 125 volts.

For units intended to be operated at 230 volts (US use): Use a UL-listed and CSA-certified cord set consisting of a minimum 18 AWG, Type SVT or SJT, three-conductor cord, a maximum of 15 feet in length and a tandem blade, grounding-type attachment plug rated 15 amperes, 250 volts.

For units intended to be operated at 230 volts (outside the US): Use a cord set with a grounding-type attachment plug. The cord set must have the appropriate safety approvals for the country in which the equipment are installed.

IBM power cords for a specific country or region are available only in that country or region.

Cautions and regulatory compliance statements for NEBS

This library is NEBS certified. This section includes the cautions and regulatory compliance statements for the Network Equipment-Building System (NEBS) certification from the Telcordia Electromagnetic Compatibility and Electrical Safety - Generic Criteria for Network Telecommunications Equipment (A Module of LSSGR, FR-64; TSGR, FR-440; and NEBSFR, FR-2063) Telcordia Technologies Generic Requirements, GR-1089-CORE, Issue 4, June 2006.

Table 1. NEBS Compliance Statements

CAUTION: To comply with the Telcordia GR-1089-CORE standard for electromagnetic compatibility and safety, for Ethernet RJ-45 ports, use only shielded Ethernet cables that are grounded on both ends. In a NEBS installation, all Ethernet ports are limited to intra-building wiring.
CAUTION: The intra-building ports of the equipment or subassembly are only suitable for connection to intra-building or unexposed wiring or cabling. The intra-building ports of the equipment or subassembly must NOT be metallically connected to interfaces that connect to the OSP or its wiring. These interfaces are designed for use only as intra-building interfaces (Type 2 or Type 4 ports as described in GR-1089-CORE, Issue 4), and require isolation from the exposed OSP cabling. The addition of primary protectors is not sufficient protection in order to connect these interfaces metallically to OSP wiring.
An external Surge Protective Device (SPD) is not required for operating this library.
This product can be installed in a network telecommunication facility or location where the NEC applies.

Glossary

This glossary defines the special terms, abbreviations, and acronyms that are used in this publication. If you do not find the term that you are looking for, refer to the index or to the *Dictionary of Computing*, 1994.

Numbers

2:1 compression

The relationship between the quantity of data that can be stored with compression as compared to the quantity of data that can be stored without compression. In 2:1 compression, twice as much data can be stored with compression as can be stored without compression.

2.5:1 compression

The relationship between the quantity of data that can be stored with compression as compared to the quantity of data that can be stored without compression. In 2.5:1 compression, two-and-a-half times as much data can be stored with compression as can be stored without compression.

3U

This library requires 3 units (3U) of rack space.

A

A

Ampere.

AC

Alternating current.

Access method

A technique for moving data between main storage and input or output devices.

Accessor
This component contains the library robot and bar code reader. The accessor moves cartridges to and from the I/O Station, storage slots, and tape drives.

Adapter card
A circuit board that adds function to a computer.

Adj
Adjustment.

Administrator (Admin)
The Admin role has access to all menus. The default password is `adm001`, and the default PIN is `0000`.

AH
Authentication Header. An Internet Protocol intended to guarantee connectionless integrity and data origin authentication of IP datagrams. Further, it can optionally protect against replay attacks by using the sliding window technique and discarding old packets.

AIX®
Advanced Interactive Executive. IBM®'s implementation of the UNIX operating system. The System p system, among others, uses AIX as its operating system.

Alphanumeric
Pertaining to a character set that contains letters, numerals, and other characters, such as punctuation marks.

Alter
To change.

Ambient temperature
The temperature of air or other media in a designated area, particularly the area that is surrounding equipment.

AME
Application Managed Encryption.

ampere (A)
A unit of measure for electric current that is equivalent to a flow of 1 coulomb per second, or to the current produced by 1 volt applied across a resistance of 1 ohm.

ANSI
American National Standards Institute.

API
Application planning interface. A set of clearly defined methods of communication between various software components.

Application-managed encryption
Tape encryption that is controlled by an application.

Archive
To collect and store files in a designated place.

ASCII
American National Standard Code for Information Interchange. A 7 bit coded character set (8 bits including parity check) that consists of control characters and graphic characters.

Assigning a device
The establishing of the relationship of a device to a running task, process, job, or program.

Assignment
The naming of a specific device to complete a function.

Asynchronous
Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals.

Attention (notice)
A word for calling attention to the possibility of danger to a program, device, or system, or to data. Contrast with *caution* and *danger*.

ATTN
Attention.

Authentication Header (AH)
A member of the IPsec protocol suite. AH guarantees connectionless integrity and data origin authentication of IP packets.

B

Backup
To make extra copies of documents or software for safekeeping.

Bar code
A code that represents characters by sets of parallel bars of varying thickness and separation, which are read optically by transverse scanning.

Bar code label
Paper bearing a bar code and having an adhesive backing. The bar code label must be affixed to a tape cartridge to enable the library to identify the cartridge and its volume serial number.

Bar code reader
A laser device that is specialized for scanning and reading bar codes and converting them into either the ASCII or EBCDIC digital character code.

Bezel
Decorative and safety cover.

Bicolored
Having two colors.

bit
Either of the digits 0 or 1 when used in the binary numbering system.

BOM or bill of materials
A list of specific types and amounts of direct materials that are expected to be used to produce a specific job or quantity of output.

Border Gateway Protocol (BGP)
BGP is the core routing protocol of the Internet. It works by maintaining a table of IP networks or 'prefixes' that designate network reachability among autonomous systems (AS).

BRMS
Backup Recovery and Media Services.

Browser
A client program that initiates requests to a web server and displays the information that the server returns.

Buffer
A routine or storage that is used to compensate for a difference in rate of flow of data or time of occurrence of events, when data is transferred from one device to another.

Bus
A facility for transferring data between several devices that are located between two end points, only one device able to transmit at a specified moment.

byte

A string that consists of some bits (usually 8) that are treated as a unit and represent a character. A fundamental data unit.

C

CA

Certificate Authority.

CA certification

In cryptography, a certificate from a certificate authority (CA).

Capacity

The amount of data that can be contained on storage media and expressed in bytes of data.

Cartridge manual rewind tool

A device that can be fitted into the reel of a cartridge and used to rewind tape into or out of the cartridge.

Cartridge memory (CM)

Within each data cartridge, an embedded electronics and interface module that can store and retrieve a cartridge's historical usage and other information.

Cartridge storage slot

Individual slot that is located within a magazine that is used to house tape cartridges.

Caution (notice)

A word to call attention to possible personal harm to people. Contrast with *attention* and *danger*.

CD

Compact Disc. A disc, usually 4.75 inches in diameter, from which data is read optically by using a laser.

CE

Customer engineer, field engineer, service representative.

Centimeter (cm)

One one-hundredth of a meter (0.01 m). Approximately 0.39 inches.

Channel command

An instruction that directs a data channel, control unit, or device to run an operation or set of operations.

Char

Character.

CHK

Check.

Cleaning cartridge

A tape cartridge that is used to clean the heads of a tape drive. Contrast with *data cartridge*.

CM

Cartridge Memory. Within each data cartridge, an embedded electronics and interface module that can store and retrieve a cartridge's historical usage and other information.

CoD

Capacity on-demand.

Command

A control signal that initiates an action or the start of a sequence of actions.

Compact disc (CD)

A disk, usually 4.75 inches in diameter, from which data is read optically by using a laser.

Compression

The process of eliminating gaps, empty fields, redundancies, and unnecessary data to shorten the length of records or blocks.

Concurrent

Refers to diagnostic procedures that can be run on one control unit while the rest of the subsystem remains available for customer applications.

Contingent connection

A connection between a channel path and a drive that is caused when a unit check occurs during an I/O operation.

Controller

A device that provides the interface between a system and one or more tape drives.

Control path drive

Controller A device that provides the interface between a system and one or more tape drives. *Control path drive* A drive that communicates messages from the host computer to the library in which the drive is installed.

Cookie

A packet of data that is exchanged between the library and a web browser to track configuration.

CP

Circuit protector.

CPF

Control Path Failover.

CRU

Customer Replaceable Unit.

CSA

Canadian Standards Association.

Ctrl

Control.

CU

Control unit.

D

Danger (notice)

A word to call attention to possible lethal harm to people. Contrast with *attention* and *caution*.

Data

Any representations such as characters or analog quantities to which meaning is or might be assigned.

Data buffer

The storage buffer in the control unit. This buffer is used to increase the data transfer rate between the control unit and the channel.

Data cartridge

A tape cartridge that is dedicated to storing data. Contrast with *cleaning cartridge*.

Data check
A synchronous or asynchronous indication of a condition that is caused by invalid data or incorrect positioning of data.

DC
Direct current.

DCS
Designated Cleaning Slot.

Degauss
Makes a magnetic tape nonmagnetic by using electrical coils that carry currents that neutralize the magnetism of the tape.

Degausser
A device that makes magnetic tape nonmagnetic.

Degradation
A decrease in quality of output or throughput or an increase in machine error rate.

Degraded
Decreased in quality of output or throughput or increased machine error rate.

Deserialize
To change from serial-by-bit to parallel-by-byte.

Detented
A part that is held in position with a catch or lever.

Device
Any hardware component or peripheral device, such as a tape drive or tape library, that can receive and send data.

Device driver
A file that contains the code that is needed to use an attached device.

DHCPv6
The Dynamic Host Configuration Protocol for IPv6. Although IPv6's stateless address autoconfiguration removes the primary motivation for DHCP in IPv4, DHCPv6 can still be used to statefully assign addresses if the network administrator wants more control over addressing.

DH group
Diffie-Hellman group.

DIAG
Diagnostic section of maintenance information manual.

Differential
See High Voltage Differential (HVD).

Direct-access storage
A storage device in which the access time is independent of the location of the data.

Display contrast
The brightness of the display on the Operator Panel.

DLL
Dynamic Link Library. The Microsoft implementation of the shared library concept. These libraries usually have the file extension dll, ocs (for libraries that contain activeX controls, or drv (for legacy system drivers).

DNS
Directory Name System. This system allows the library to recognize text-based addresses instead of numeric IP addresses.

Download
To transfer programs or data from a computer to a connected device, typically a personal computer.
To transfer data from a computer to a connected device, such as a workstation or personal computer.

DPF
Data Path Failover.

DRAM
Dynamic random-access memory.

Drive, magnetic tape
A mechanism for moving magnetic tape and controlling its movement.

Drive Not Configured
This message occurs during the first boot after a factory settings restore is run. This message is not a real issue since it takes time for the library to configure.

DRV
Drive.

DSA key
Encryption key type.

DSE
Data security erase.

DSP
Digital signal processor.

E

EBCDIC
Extended binary-coded decimal interchange code.

EC
Edge connector. Engineering change.

ECC
Error correction code.

EEB
Ethernet Expansion Blade.

EEPROM
Electrically erasable programmable read-only memory.

EIA
Electronics Industries Association.

EIA unit
A unit of measure, which is established by the Electronic Industries Association, equal to 44.45 millimeters (1.75 inches).

Eject
To remove or force out from within.

EKM

Encryption Key Manager.

Electronic mail
Correspondence in the form of messages that are transmitted between user terminals over a computer network.

Email
See *electronic mail*.

Encryption
A method of storing data in a format that helps protect data from inadvertent or deliberate compromise. An encryption-enabled drive contains the necessary hardware and firmware to encrypt and decrypt host tape application data. Encryption policy and encryption keys are provided by the host application or host server.

Encryption key manager (EKM)
A software program that assists IBM-encrypting tape drives in generating, protecting, storing, and maintaining encryption keys that encrypt information that is written to and decrypt information that is read from tape media.

Entitlement
IBM Entitlement is the official right to receive service and support for your tape library.

EPO
Emergency power off.

EPROM
Erasable programmable read only memory.

EQC
Equipment check.

Equipment check
An asynchronous indication of a malfunction.

Error log
A data set or file in a product or system where error information is stored for later access.

ESD
Electrostatic discharge.

ESP
Encapsulating Security Payload. An Internet Protocol that provides origin authenticity, integrity, and confidentiality protection of a packet. ESP also supports encryption-only and authentication-only configurations, but encryption without authentication is discouraged because it is insecure.

F

FAT32
FAT stands for File Allocation Table. FAT32 is an extension which means that data is stored in chunks of 32 bits. Any USB flash drive that is used for updating firmware or exporting logs for the TS4300 library must be in this format.

Fault symptom code (FSC)
A hexadecimal code that is generated by the drive or the control unit microcode in response to a detected subsystem error.

FC
Fibre Channel, Feature code.

FCC
Federal communications commission.

FE
Field engineer, customer engineer, or service representative.

FH
Full Height.

Fibre Channel
A high-speed method to connect data storage to a server. The British spelling of "Fibre" is used because the technology can be used with either fiber optic or copper cables. Thus, the name does not imply that it can be used only with a fiber optic cable.

Fiducial
A target that is used for teaching a physical location to a robot.

Field replaceable unit (FRU)
An assembly that is replaced in its entirety when any one of its components fails.

File
A named set of records that are stored or processed as a unit. Also referred to as a data set.

File protection
The processes and procedures that are established in an information system that are designed to inhibit unauthorized access to, contamination of, or deletion of a file.

File transfer protocol (FTP)
In the Internet suite of protocols, an application layer protocol that uses TCP and Telnet services to transfer bulk-data files between machines or hosts.

Firmware
Proprietary code that is delivered as microcode as part of an operating system. Firmware is more efficient than software loaded from an alterable medium and more adaptable to change than pure hardware circuitry. An example of firmware is the Basic input/output system (BIOS) in read-only memory (ROM) on a PC system board.

FLASH EEPROM
An electrically erasable programmable read-only memory (EEPROM) that can be updated.

FMR
Field microcode replacement.

Format
The arrangement or layout of data on a data medium.

Formatter
Part of a magnetic tape subsystem that runs data conversion, speed matching, encoding, first-level error recovery, and interfaces to one or more tape drives.

FP
File protect.

Frayed
Damaged as if by an abrasive substance.

FRU
Field replaceable unit.

FSC
Fault symptom code.

FSI

Fault symptom index.

FTSS Field Technical Sales Support.

Functional microcode
Microcode that is resident in the machine during normal customer operation.

G

g Gram.

GB gigabyte.

GBIC Gigabit Interface Converter.

Gb/s gigabits/second

Gbit gigabit

gigabit (Gbit)
1 000 000 000 bits.

gigabyte (GB)
1 000 000 000 bytes.

Gigabit Interface Converter (GBIC)
Converts copper interface to optic interface.

Gnd Ground.

GUI Graphical User Interface

H

HBA Host Bus Adapter.

HD Slot Technology
High-density (HD) slot technology. Allows multiple cartridges to be stored in a tiered architecture.

hertz (Hz)
Unit of frequency. 1 hertz equals one cycle per second.

hex Hexadecimal.

HH Half Height.

High Voltage Differential (HVD)
A logic-signaling system that enables data communication between a supported host and the library. HVD signaling uses a paired plus and minus signal level to reduce the effects of noise on the SCSI bus. Any noise that is injected into the signal is present in both a plus and minus state, and is canceled. Synonymous with *differential*.

HVD SCSI Bus High-Voltage Differential.

Hz Hertz (cycles per second).

I

IBM Security Key Lifecycle Manager (SKLM)
IBM's EKM application that assists encrypting tape drives in generating, protecting, storing, and maintaining encryption keys that encrypt information that is written to and decrypt information that is read from tape media.

IBM Spectrum Archive
Formerly known as Linear Tape File System (LTFS). A file system that works with LTO Generation tape technology to access data stored on an IBM tape cartridge.

IBM Ultrium Tape Drive
Located within the library, a data-storage device that controls the movement of the magnetic tape in an IBM LTO Ultrium Tape Cartridge. The drive houses the mechanism (drive head) that reads and writes data to the tape.

ID Identifier.

Identifier (ID)
(1) In programming languages, a lexical unit that names a language object. For example, the names of variables, arrays, records, labels, or procedures. An identifier usually consists of a letter optionally followed by letters, digits, or other characters. (2) One or more characters that are used to identify or name data element and possibly to indicate certain properties of that data element. (3) A sequence of bits or characters that identifies a program, device, or system to another program, device, or system.

IEC International Electrotechnical Commission.

IKE Internet Key Exchange that is used in the IPSec protocol.

IML Initial microprogram load.

Incompatible magazine
This message might display on the Operator Panel during library initialization. It occurs during factory restore or VPD. This message is not a real issue since it takes time for the library to configure.

Initial microprogram load (IML)

The action of loading a microprogram from external storage to writable control storage.

Initiator

The component that runs a command. The initiator can be the host system or the tape control unit.

INST

Installation.

Interface

A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

Internet Key Exchange (IKE)

The protocol that is used to set up a security association (SA) in the IPSec protocol suite. See also *Security Association (SA)*.

Internet Protocol Version 4 (IPv4)

See *IPv4*.

Internet Protocol Version 6 (IPv6)

See *IPv6*.

Interposer

The part that is used to convert a 68-pin connector to a 50-pin D-shell connector.

Intervention required

Manual action is needed.

INTRO

Introduction.

I/O

Input/output.

I/O Station

Cartridge location that is dedicated for the insertion of cartridges into and the removal of cartridges from the library.

IOP

Input/output processor.

IP

Internet Protocol.

IP address

An identifier for a computer or device on an Internet Protocol (TCP/IP) network. Networks that use the TCP/IP protocol route messages that are based on the IP address of the destination. See *IPv4* and *IPv6*.

IPL

Initial program load.

IPSec (IP Security)

A set of protocols for securing IPv6 network communications by authentication and encryption.

IP Stack

A TCP/IP protocol stack that manages static IP addresses.

IPv4

A network layer protocol for packet-switched networks. IPv4 supports 2^{32} (about 4.3 billion) addresses.

IPv6

A network layer protocol for packet-switched networks. It is the designated successor of IPv4 for general use on the Internet. The main improvement of IPv6 is the increase in the number of addresses available for networked devices, allowing, for example, each mobile phone and mobile electronic device to have its own unique address.

ISV

Independent software vendor.

ITDT

IBM Tape Diagnostic tool.

ITST

Idle-time self-test.

K

Kerberos

Kerberos Authentication is a standard (RFC 1510) third-party authentication protocol that provides end-to-end security for distributed computing environments.

Key Path Diagnostics (KPD)

Key Path Diagnostics is a test tool that provides details to troubleshoot Encryption communication issues.

kilogram (kg)

1000 grams (approximately 2.2 pounds).

km

kilometer. 1000 Meters, Approximately 5/8 mile.

KMIP

Key Management Interoperability Protocol.

L

LAN

Local area network. A computer network within a limited area.

LCB

Library Control Blade.

LCD

See *liquid crystal display*.

LDAP

Lightweight Directory Access Protocol. This protocol allows the library to use login and password information that is stored on a server to grant access to the library functions.

LDAPS

Secure LDAP over SSL.

LDI

Library Drive Interface.

LED
Light-emitting diode.

Library certification
In cryptography, a certificate that is provided by the library.

Library-managed encryption
Tape encryption that is controlled by the tape library.

Linear Tape-Open (LTO)
A type of tape storage technology that is developed by the IBM Corporation, Hewlett-Packard, and Quantum. LTO technology is an "open format" technology, which means that its users have multiple sources of product and media. The "open" nature of LTO technology enables compatibility between different vendors' offerings by ensuring that vendors comply with verification standards. The LTO technology is implemented in two formats: the Accelis format focuses on fast access; the Ultrium format focuses on high capacity. The Ultrium format is the preferred format when capacity (rather than fast access) is the key storage consideration. An Ultrium cartridge has a compressed data capacity of up to 30 TB (2.5:1 compression) and a native data capacity of up to 12 TB.

Liquid crystal display (LCD)
A low-power display technology that is used in computers and other I/O devices.

Loadable
The ability to be loaded.

LME
Library Managed Encryption.

LTO
See *Linear Tape-Open*.

LTO cartridge memory (LTO-CM)
Within each LTO Ultrium data cartridge, an embedded electronics and interface module that can store and retrieve a cartridge's historical usage and other information.

LUN
Logical Unit Number.

LVD
SCSI Bus Low Voltage Differential.

M

M8
LTO 8 Type M Cartridge.

MAC address
The Media Access Control address of a computer networking device.

Magnetic tape
A tape with a magnetic surface layer on which data can be stored by magnetic recording.

Management GUI
Web User Interface, Web UI, Web GUI.

MAP
Maintenance analysis procedure.

Mask
A pattern of characters that controls the retention or elimination of portions of another pattern of characters. To use a pattern of characters to control the retention or elimination of portions of another pattern of characters.

Master file
A file that is used as an authority in a job and that is relatively permanent, even though its contents might change. Synonymous with main file.

Maximum Transmission Unit (MTU)
The size of the largest packet that a network protocol can transmit.

MB
Megabyte (expressed as data rate in MB/s or MB/second).

Media capacity
The amount of data that can be contained on a storage medium, expressed in bytes of data.

Media-type identifier
Pertaining to the bar code on the bar code label of the IBM Ultrium tape cartridge, a two-character code, L1, that represents information about the cartridge. **L** identifies the cartridge as one that can be read by devices that incorporate LTO technology; **1** indicates that it is the first generation of its type.

Mega
One million of.

meter
In the Metric System, the basic unit of length equal to approximately 39.37 inches.

MIB
Management Information Base. Information repository that is used by SNMP.

Micro
One millionth of.

Microcode
(1) One or more micro instructions. (2) A code, representing the instructions of an instruction set, which is implemented in a part of storage that is not program-addressable. (3) To design, write, and test one or more micro instructions. (4) See also *microprogram*.

Microdiagnostic routine
A program that runs under the control of a supervisor, usually to identify field replaceable units.

Microdiagnostic utility
A program that is run by the customer engineer to test the machine.

Microinstruction
A basic or elementary machine instruction.

Microprogram
A group of micro instructions that when run completes a planned function.
The term microprogram represents a dynamic arrangement or selection of one or more groups of micro instructions for execution to complete a particular function.
The term microcode represents microinstructions that are used in a product as an alternative to hard-wired circuitry to implement certain functions of a processor or other system component.

MIM
Media information message.

mm	Millimeter.
Modifier	That which changes the meaning.
Monitor	The Monitor role is an interchangeable term corresponding to the User role. The Monitor role has viewing privileges to the unit, but is not able to make configuration changes.
Mount a device	To assign an I/O device with a request to the operator.
MP	Microprocessor.
ms	Millisecond.
MSG	Message.
Multipath	Pertaining to using more than one path.

N

N/A	Not applicable.
Network Address Translation (NAT)	NAT involves rewriting the source or destination addresses of IP packets as they pass through a router or firewall. Most systems that use NAT do so to enable multiple hosts on a private network to access the Internet over a single public IP address.
NEMA	National Electrical Manufacturers Association.
Node	In a network, a point at which one or more functional units connect channels or data circuits.
NTFS	New Technology File System. The primary file system that is used in Windows.
NTP	Network Time Protocol. This protocol allows the library to set its internal date and time that is based on the date and time of a server.
NVS	Nonvolatile storage. A storage device whose contents are not lost when power is cut off.

O

OCP	Operator Panel (Operator Control Panel).
Oersted	The unit of magnetic field strength in the unrationalized centimeter-gram-second (cgs) electromagnetic system. The oersted is the magnetic field strength in the interior of an elongated, uniformly wound solenoid that is excited with a linear current density in its winding of 1 ampere per 4 π centimeters of axial length.
Offline	Pertaining to the operation of a functional unit without the continual control of a computer. Contrast with <i>online</i> .
Online	Pertaining to the operation of a functional unit that is under the continual control of a computer. Contrast with <i>offline</i> .
OPER	Operation.
OV	Over voltage.
Overrun	Loss of data because a receiving device is unable to accept data at the rate it is transmitted.
Overtightening	To tighten too much.

P

Parameter	A variable that is given a constant value for a specified application and that might denote the application.
p bit	Parity bit.
PC	Parity check.
PCC	Power® control compartment.
PDF	Portable Document Format.
PE	Parity error. Product engineer.
PFS	Perfect forward secrecy.
Pick	Pertaining to the library to remove, by using a robotic device, a tape cartridge from a storage slot or drive.
Picker	A robotic mechanism that is located inside the library that moves cartridges between the cartridge storage slots and the drive.
PM	

	Preventive maintenance.
POR	Power-on reset.
Port	A physical connection for communication between the 3590 and the host processor. The 3590 has 2 SCSI ports.
Portable Document Format (PDF)	A standard that is specified by Adobe Systems, Incorporated, for the electronic distribution of documents. PDF files are compact, can be distributed globally (by way of email, the web, intranets, or CD-ROM), and can be viewed with the Acrobat Reader. Acrobat Reader is software from Adobe Systems that can be downloaded at no cost from the Adobe Systems home page.
Private key	A cryptographic key that is used to decrypt a message.
PROM	Programmable read only memory.
PS	Power supply.
PTF	Program temporary fix. A single bugfix or group of bugfixes that are distributed in a form ready to install for customers.
PWR	Power.

R

Rack	A unit that houses the components of a storage subsystem, such as the library.
Rackmount kit	A packaged collection of articles that are used to install the rack-mounted version of the library.
RAM	Random access memory.
Random access memory	A storage device into which data is entered and from which data is retrieved in a nonsequential manner.
Random Mode	In Random mode, the library allows the server's (host's) application software to select any data cartridge in any order.
RAS	Reliability, availability, and serviceability.
Record	A collection of related data or words, which are treated as a unit.
Recording density	The number of bits in a single linear track measured per unit of length of the recording medium.
Recoverable error	An error condition that allows continued execution of a program.
Ref	Reference.
Reg	Register.
Reinventory	To inventory again.
REST	Representational state transfer. Part of an API. REST systems aim for fast performance, reliability, and the ability to grow, by reusing components that can be managed and updated without affecting the system as a whole, even while it is running.
Retension	The process or function of tightening the tape onto the cartridge, if it is sensed that the tape has a loose wrap on the cartridge.
RFC (Request for Comments)	Request for Comments (RFC) documents are a series of memoranda, which encompasses new research, innovations, and methodologies applicable to Internet technologies.
RH	Relative humidity.
RID tag	Repair identification tag.
RML	Rack Mount Line.
Robot	Picker.
Robotic Assembly	The picker, picker assembly.
Robotics	Picker assembly.
Root CA certification	In cryptography, a root certificate from a certificate authority (CA).
RPQ	Request for price quotation.
RSA key	Encryption key type.
R/W	read/write.

S

	Seconds of time.
SAC	Service Action Code. Code that is developed to indicate possible FRU or FRUs to replace to repair the hardware.
SAN	Storage area network.
SAS	Serial Attached SCSI. A computer bus technology and serial communication protocol for direct attached storage devices. SAS is a replacement for parallel SCSI with higher speeds, but still uses SCSI commands.
Scratch cartridge	A data cartridge that contains no useful data, but can be written to with new data.
SCD	Single Character Display.
SCSI	Small computer system interface.
SE	Single-ended.
Sequential Mode	Sequential Mode is intended to be used by host applications that aren't supporting SCSI media changer devices but need to get another cartridge loaded if the current cartridge is full.
Segment	A part.
Sel	Select.
Serial Attached SCSI (SAS)	A drive with a SAS interface can be linked directly to controllers. SAS is a performance improvement over traditional SCSI because SAS enables multiple devices (up to 128) of different sizes and types to be connected simultaneously with thinner and longer cables. It supports full-duplex signal transmission up to 3 Gb/s. In addition, SAS drives can be hot-plugged.
Serialize	To change from parallel-by-byte to serial-by-bit.
Serializer	A device that converts a space distribution of simultaneous states, which represents data into a corresponding time sequence of states.
Service	Access to this level is for Service personnel only - Service personnel have access to all menus.
Servo, servos	An adjective for use in qualifying some part or aspect of a servomechanism.
Servomechanism	A feedback control system in which at least one of the system signals represents mechanical motion.
Signature	A digital signature that is used in cryptography to identify one party to ensure authenticity.
SKLM (IBM Security Key Lifecycle Manager)	IBM EKM application that assists encrypting tape drives in generating, protecting, storing, and maintaining encryption keys that encrypt information that is written to and decrypt information that is read from tape media.
SKLM for z/OS®	Security Key Lifecycle Manager for IBM System z® mainframes.
Slot blocker	A slot blocker is used to restrict/close off a data cell so a data cartridge cannot be inserted.
Small Computer Systems Interface (SCSI)	A standard that is used by computer manufacturers for attaching peripheral devices (such as tape drives, hard disks, CD-ROM players, printers, and scanners) to computers (servers). Pronounced "scuzzy". Variations of the SCSI interface provide for faster data transmission rates than standard serial and parallel ports (up to 320 megabytes per second). The variations include • Fast/Wide SCSI - Uses a 16-bit bus, and supports data rates of up to 20 MBps. • SCSI-1 - Uses an 8-bit bus, and supports data rates of 4 MBps. • SCSI-2 - Same as SCSI-1, but uses a 50-pin connector instead of a 25-pin connector, and supports multiple devices. • Ultra-SCSI - Uses an 8- or 16-bit bus, and supports data rates of 20 or 40 MBps. • Ultra2 SCSI - Uses an 8- or 16-bit bus and supports data rates of 40 or 80 MBps. • Ultra3 SCSI - Uses a 16-bit bus and supports data rates of 80 or 160 MBps. • Ultra160 SCSI - Uses a 16-bit bus and supports data rates of 80 or 160 MBps. • Ultra320 SCSI - Uses a 16-bit bus and supports data rates of 320 MBps.
SMI-S	See <i>Storage Management Initiative Specification (SMI-S)</i> .
SMTP	Simple Mail Transfer Protocol. SMTP is a standard for email transmissions across the internet.
SNMP	Simple Network Management Protocol. SNMP is used by network management systems to monitor network-attached devices for conditions that warrant administrative attention.
SNTP	Simple Network Time Protocol. Used to synchronize the clocks of network-attached devices.
SMW	Servo Manufacturer's Word.
SNS	Sense.
Special feature	A feature that can be ordered to enhance the capability, storage capacity, or performance of a product, but is not essential for its basic work.
SPI	Security Parameters Index.
SR	Service representative, see also <i>CE</i> .
SRAM	

	Static random access memory.
SS	Status store.
SSH	Secure Shell.
SSL (Secure Sockets Layer)	A set of cryptographic protocols for secure communications on the Internet for such things as web browsing, email, Internet faxing, instant messaging, and other data transfer. SSL allows applications to communicate across a network in a way that is designed to prevent eavesdropping, tampering, and message forgery.
SSP	Serial SCSI Protocol.
ST	Store.
Standard feature	The significant design elements of a product that are included as part of the fundamental product.
START	Start maintenance.
StartTLS	Secure LDAP communication that uses TLS.
Storage Management Initiative Specification (SMI-S)	A storage standard that is developed and maintained by the Storage Networking Industry Association (SNIA). It is also ratified as an ISO standard. The main objective of SMI-S is to enable broad interoperable management of heterogeneous storage vendor systems.
Subsystem	A secondary or subordinate system, capable of operating independently of, or asynchronously with, a controlling system.
Superuser	The Superuser role has access to most sections of the library menus.
SUPP	Support.
Sync	Synchronous, synchronize. Occurring with a regular or predictable time relationship.

T

Tachometer, tach	A device that emits pulses that are used to measure/check speed or distance.
Tape cartridge	A container that holds magnetic tape that can be processed without separating it from the container.
Tape void	An area in the tape in which no signal can be detected.
TB	Terabyte.
TCP/IP	Transmission Control Protocol/Internet Protocol.
TCU	Tape control unit.
Terabyte	One terabyte = 1,000,000,000,000 bytes, or 1000 gigabytes (GBs).
TH	Thermal.
TKLM (IBM Tivoli® Key Lifecycle Manager)	IBM's EKM application that assists encrypting tape drives in generating, protecting, storing, and maintaining encryption keys that encrypt information that is written to and decrypt information that is read from tape media.
thread/load operation	A procedure that places tape along the tape path.
TLS	Transport :Layer Security.
TM	Tapemark, Trademark.
Transport mode	End-to-end communications security in which the end-point computers do the security processing.
Trusted certification	In cryptography, a trustworthy certificate that is not registered with a certificate authority.
Tunnel mode	Port-to-port communications security in which security is provided to several machines by a single node.

U

UART	Universal asynchronous receiver/transmitter.
UID	Unit Identification.
UL	Underwriter's Laboratories.
Universal rack connector	A rack-mount kit has four universal rack connectors as part of the kit. Each connector has two sides - one side is for round-hole racks, and the other side is for square-hole racks. The square-hole side might be painted. The connectors are installed from the inside of the rack out, and the rails are hooked onto them. See Figure 1 .
Unload	Prepare the tape cartridge for removal from the drive.

User	The User role is an interchangeable term corresponding to the Monitor role. The User role has viewing privileges to the unit, but is not able to make configuration changes.
Utilities	Utility programs.
Utility programs	A computer program in general support of the processes of a computer. For instance, a diagnostic program.
UV	Under voltage.

V

VOLSER	Volume serial number.
Volume	A certain portion of data, together with its data carrier, that can be handled conveniently as a unit.
VPD	Vital product data. The information that is contained within the tape drive that requires nonvolatile storage that is used by functional areas of the drive, and information that is required for manufacturing, RAS, and engineering.

W

Web UI, Web GUI, Web User Interface	Management GUI
Word	A character string that is convenient for some purpose to consider as an entity.
Worldwide Node Name (WWNN)	A unique character string that identifies Fibre Channel Host Bus adapters (HBA).
WORM	Write Once, Read Many.
Write	Write command.
WT	World trade.
WWCID	Worldwide Cartridge Identifier.
WWN	Worldwide Name.
WWNN	Worldwide Node Name.
WWPN	Worldwide port name.

X

XR	External register.
XRA	External register address register.

Publications

The Publications section provides instructions to print selected IBM Documentation topics, and can help you locate other publications that are related to the libraries and drives.

Complete these steps to print one or more IBM Documentation topics.

Note: The machine that is displaying the IBM Documentation must be connected to a printer, or must have PDF print software that is installed to print IBM Documentation topics.

1. In the left navigation pane of the IBM Documentation, hover over or highlight the topic that you want to print.
2. Click or select the Open Quick Menu icon that is displayed to the right of the topic title. This icon is a gray circle that contains a downward-pointing arrow. The Quick Menu is displayed.
3. To print just the active topic, click or select Print this topic. To print the active topic and all its subtopics, click or select Print this topic and subtopics.
Note: The maximum number of pages that can be printed at one time is 100.
4. The Print Preview window opens, displaying the topics to be printed. Scroll to the end of this preview to ensure that all topics display.
5. Select Print from the File drop-down menu.
6. The Print dialog box opens. Use the Name drop-down menu to select the name of the printer you want to use to print the IBM Documentation topics.
7. Select the All radio button in the Print Range box.
8. Select the selected frame radio button in the Print Frames box.
9. Click OK.

Use the following links to locate other publications that are related to the TS4300 drives.

- [Related publications](#)
This topic lists sources that provide additional information about the IBM® TS4300 Tape Library and its associated products.

Related publications

This topic lists sources that provide additional information about the IBM® TS4300 Tape Library and its associated products.

Note: The IBM TS4300 Tape Library is a customer installed unit. The customer is responsible for the setup and maintenance of the tape library. The customer is charged for service if a service contract isn't in place.

Related Publications

Refer to the following publications for information. To ensure that you have the current publications, go to <https://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.

- [IBM TS4300 Tape Library Getting Started Guide](#) (SC27-4630) provides installation information.
- [IBM TS4300 Tape Library User's Guide](#) (SC27-4629) provides planning, installation, and maintenance information.
- [IBM Tape Device Drivers and Diagnostic Tool User's Guide](#) provides instructions for attaching IBM-supported hardware to open-systems operating systems. It indicates what devices and levels of operating systems are supported. It also gives requirements for adapters, and tells how to configure hosts to use the device driver. All of the above are with the Ultrium family of devices.
- [IBM Tape Device Drivers Programming Reference](#) supplies information to application owners who want to integrate their open-systems applications with IBM-supported Ultrium hardware. The reference contains information about the application programming interfaces (APIs) for each of the various supported operating-system environments.
- *IBM Environmental Notices and User Guide (ENUG)* (z125-5823), which is at https://www.ibm.com/support/knowledgecenter/ENVSAF_SHR/envsafetynotice/envsafetynotice_kickoff.htm. This information includes statements on limitations, product information, product recycling and disposal, battery information, flat panel display, refrigeration and water-cooling systems, external power supplies, and safety data sheets.
- *IBM Security Key Lifecycle Manager Knowledge Center*, which is at <http://www-01.ibm.com/support/knowledgecenter/SSWPVP/welcome?lang=en>, contains information to help you install, configure, and use the IBM Security Key Lifecycle Manager.
- *IBM Safety Notices*, (G229-9054) at <https://www-01.ibm.com/support/docview.wss?uid=isg27d40fbeb5e10ceb985256e31007281fa>.
- The IBM Publications Center at <http://www.ibm.com/shop/publications/order>. The Publications Center is a worldwide central repository for IBM product publications and marketing material with a catalog of 70,000 items. Extensive search facilities are provided. Payment options for orders are with credit card (in the US) or customer number for 20 countries. Many publications are available online in various file formats, and they can all be downloaded by all countries, free of charge.

IBM Tape Device Drivers and Diagnostic Tool User's Guide

Welcome to the IBM Tape Device Drivers and Diagnostic Tool User's Guide documentation, where you can find information about how to install, maintain, and use IBM Tape Device Drivers and the IBM Tape Diagnostic Tool (ITDT) on multiple platforms.

Last updated: 2022-03-22.

- [Introduction](#)
The IBM® tape and medium changer device drivers are designed specifically to take advantage of the features that are provided by the IBM 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 unmount), and 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. IBM maintains the levels of device drivers and driver documentation for the drive on the Internet. You can access this material at <http://www.ibm.com/support/fixcentral>.
- [Common extended features](#)
- [AIX Tape and Medium Changer device driver](#)
- [Linux Tape and Medium Changer device driver](#)
- [Windows Tape and Medium Changer device driver](#)
- [IBM Tape Diagnostic Tool \(ITDT\)](#)
- [Verifying correct attachment of your devices](#)
- [Managing the microcode on the IBM tape drive](#)
- [Notices](#)
- [Terms and conditions for knowledge centers](#)

Permissions for the use of these publications are granted subject to the following terms and conditions.

Installation and User's Guide

- [Introduction](#)
- [Common extended features](#)
- [AIX Tape and Medium Changer device driver](#)
- [Linux Tape and Medium Changer device driver](#)
- [Windows Tape and Medium Changer device driver](#)
- [IBM Tape Diagnostic Tool \(ITDT\)](#)

Troubleshooting and support

- [Verifying correct attachment of your devices](#)
- [Managing the microcode on the IBM tape drive](#)
- [IBM tape storage support](#)
- [IBM Support home page](#)

More information

- [IBM Tape Device Drivers Programming Reference](#)

- [IBM Tape Device Drivers Installation and User's Guide \(Legacy\)](#)
- [IBM TS4500 tape library documentation](#)
- [IBM TS4300 tape library documentation](#)
- [IBM TS2900 tape autoloader documentation](#)
- [IBM TS2270 tape drive documentation](#)
- [IBM TS2280 tape drive documentation](#)
- [IBM Community \(Community platform\)](#)
- [IBM Support content \(Product support\)](#)
- [Redbooks home page](#)

©Copyright IBM Corporation 2017-2021. Last updated: 2022-03-22.

Introduction

The IBM® tape and medium changer device drivers are designed specifically to take advantage of the features that are provided by the IBM 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 unmount), and 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. IBM maintains the levels of device drivers and driver documentation for the drive on the Internet. You can access this material at <http://www.ibm.com/support/fixcentral>.

Hardware requirements

The tape drivers and the IBM Tape Diagnostic Tool (ITDT) are developed to support various versions of different platforms. For the latest support, refer to the Interoperation Center website - <http://www.ibm.com/systems/support/storage/config/ssic/>.

Note: 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 either tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems are observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and by using SAN zoning to minimize contention. IBM is focused on assuring server/storage configuration interoperability. It is strongly recommended that your overall implementation plan includes provisions for separating disk and tape workloads.

For information about this issue, see the following Redbook, section 4.1.3 in <http://www.redbooks.ibm.com/abstracts/sg246502.html?Open>.

Software requirements

Important: If you use a third-party application, consult with your application provider about the compatibility with IBM tape device drivers.

Industry-leading compatible software offerings provide storage and tape management software for the LTO tape drives. Supporting software and applications must be obtained separately from IBM®, IBM Business Partners, or independent software vendors (ISV). For a list of compatible software and additional information, refer to the ISV Matrix that is available at [Independent Software Vendor \(ISV\) matrix for 3592 and LTO](#).

IBM tape products

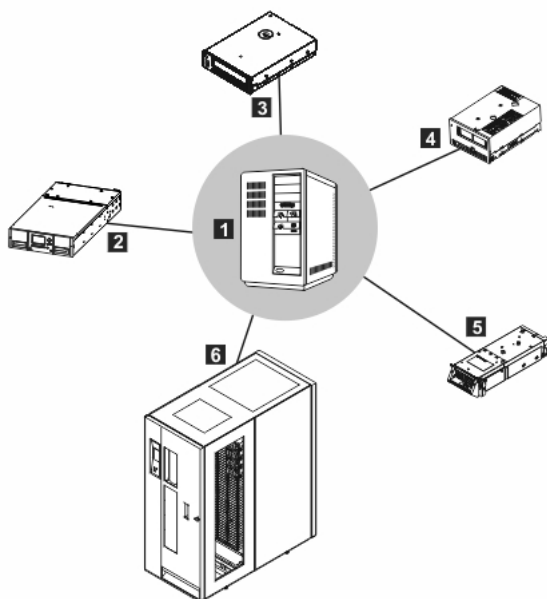
The IBM tape product family provides an excellent solution for customers with small to large storage and performance requirements.

Remember: xx in the following list represents generation of the drive.

- IBM TS22xx Tape Drive
- IBM TS2350/TS2360 Tape Drive
- IBM TS11xx Tape Drive (Enterprise)
- IBM TS3500 and TS4500 Tape Library (also known as IBM UltraScalable tape library 3584)
- IBM TS4300 Tape Library
- IBM TS2900 Tape Autoloader

The image illustrates the attachment of various current products to an open systems server.

Figure 1. Current attachment array



1 Open Systems Server	4 IBM TS2350/TS2360 (or 3580) Tape Drive
2 IBM TS4300 Tape Library	5 IBM TS11xx Tape Drive [also known as Enterprise]
3 IBM TS22xx Tape Drive	6 IBM TS3500 or TS4500 tape library

Common extended features

Purpose

This chapter provides general information about the IBM® device drivers, requirements, and advanced functionality.

- [Path failover and load balancing](#)
- [Dynamic Runtime Attributes](#)
- [Data encryption](#)
- [Recommended access order \(RAO\) open function](#)

RAO was first introduced on IBM enterprise tape drives in the 3592-E07 (TS1140). RAO enables tape control applications to accelerate the retrieval of a certain number of files from a single tape thereby reducing the seek time between those files.

Path failover and load balancing

Device driver path failover support configures 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 tape devices by using multiple host bus adapters (HBA).

Path failover is supported on certain tape products with the latest IBM® device drivers available on the following website - <http://www.ibm.com/support/fixcentral>. Instructions for downloading drivers can be found in [Accessing documentation and software online](#). Some devices require a path failover feature code to be installed before path failover support is enabled in the device driver. Refer to [Supported devices and feature codes](#) for a list of supported devices and what path failover feature code is required for your machine type.

At startup or configuration, the system detects multiple logical devices of the tape drive. 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 because they represent the same physical device.

Without path failover support, if 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, but the application must be restarted from the beginning. A long backup or restore operation might be in progress when the path error occurred. Sometimes manual failover requires operator intervention to reset the drive because a SCSI Reservation might still exist on the failing HBA path.

When path failover support is enabled on all logical devices, 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. If an application opens the primary device and a permanent path error occurs, the device driver initiates failover error recovery automatically on an alternate path. If successful, the current operation continues on an alternate path without interrupting the application. The data path failover error recovery first restores the previous device state, SCSI Reservation, and tape position. Then, it tries the failing operation again.

- [Supported devices and feature codes](#)
- [Automatic failover](#)
- [Dynamic load balancing](#)

Supported devices and feature codes

Path failover is supported only for the devices that are listed in [Table 1](#). Path failover includes Control Path failover (CPF) for tape libraries and Data Path failover (DPF) for tape drives. To use path failover support, some devices require feature codes as listed in [Table 1](#).

Table 1. Supported devices and feature codes

Supported tape library/driver	Feature code (FC), if required
TS3500 and TS4500	Standard, no FC required (DPF) FC 1682 (CPF)
TS4300/LTO	FC 1682 (CPF and DPF)

Notes:

1. Path failover is only supported on SAS devices that are attached to Windows and Linux® for Intel/AMD processor-based servers. SAS is not supported on System p servers (AIX® and Linux).
2. If your device does not support path failover, you must disable this option in the device driver. See the specific platform section for driver default behavior and enable/disable failover instructions.

Automatic failover

The automatic failover support provides error recovery on an alternate path when a permanent error occurs on the primary path. This support is transparent to the running application. The two types of path failover are Data Path failover (DPF) and Control Path failover (CPF). They are closely related. However, the difference is that DPF is an automatic failover support for the transfer of data, which provides error recovery for systems that are connected to tape drives. CPF is an automatic failover support for the transfer of commands to move tape cartridges. Examples of different configurations that can be constructed follow.

- [Data path failover](#)
- [Control path failover](#)

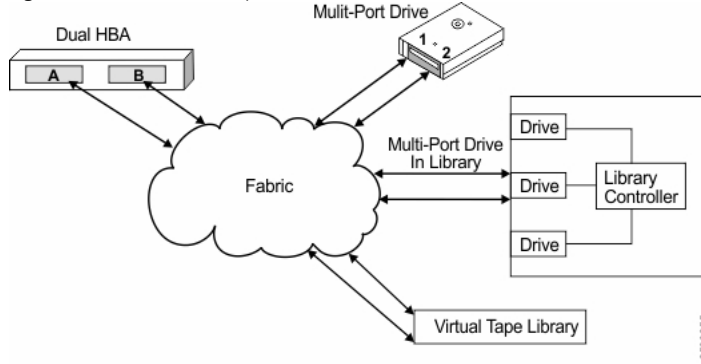
Data path failover

The following flowcharts outline the different types of configurations for data path failover (DPF). These configurations are presented in order of best practices as recommended by IBM.

Dual Host Bus Adapters (HBAs) to a multi-port drive

Consider a multipath connection that consists of two Host Bus Adapters (HBAs) connected through a fabric to a multi-port drive.

Figure 1. Dual HBA and multi-port drives



As seen in [Figure 1](#), four available paths are available between the drive and the host system. These paths are

- HBA A to drive port 1 [A, p1]
- HBA A to drive port 2 [A, p2]
- HBA B to drive port 1 [B, p1]
- HBA B to drive port 2 [B, p2]

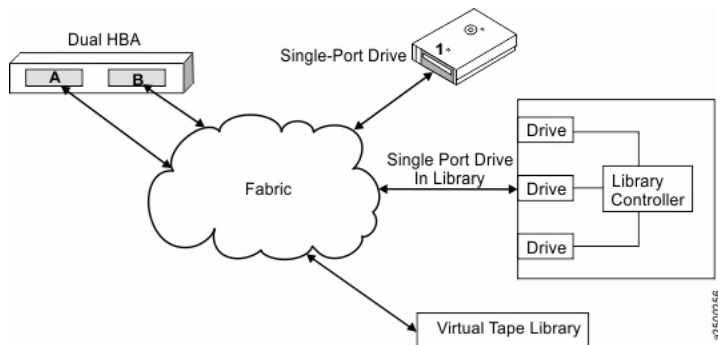
One path is the primary path and the other three are alternate paths. If [A, p1] is the primary path and if HBA A fails, two valid paths ([B, p1] and [B, p2]) remain. The DPF tries to switch to one of the available configured paths. Conversely, if the cable to port 1 of the drive fails with [A, p1] as the primary path, two valid paths to the drive [A, p2] and [B, p2] are still available. Without DPF support, if a permanent path error occurs (because of HBA or cable failover, for example), the application fails. With DPF, if the permanent failure occurs with this configuration, two valid physical paths for the data are still available for transmitting data. The running application is not affected.

If the path that failed is restored, the device driver claims the path as available and uses it as a valid alternate path in most conditions. This action is dependent on Operating System and HBA behavior, not the IBM® tape device driver behavior.

Dual Host Bus Adapters (HBAs) to a single-port drive

Consider a multipath connection that consists of two HBAs connected through a fabric to a single-port device.

Figure 2. Dual HBA and single-port drive

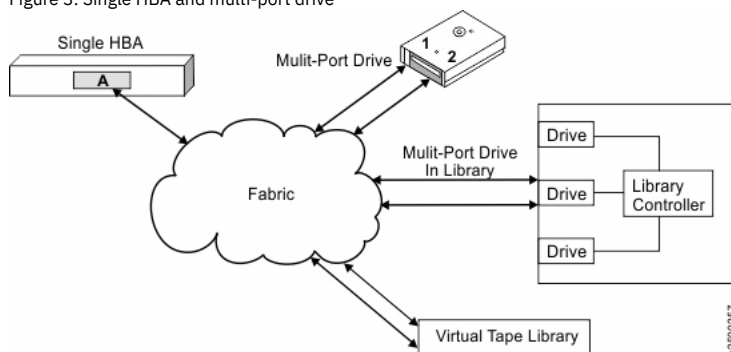


This configuration supplies two physical paths to the same device. However, if the port or cable from the device fails, the automatic failover does not work. That connection is severed and a permanent path error occurs. If, however, the failure was with one of the HBAs or their cables, the automatic data path failover selects the other HBA. Then, the information continues through the alternate path. An example here is with the connections [A, p1] and [B, p1]. If [A, p1] is the primary path and a failure occurs with the HBA or HBA cable, then DPF automatically moves the connection to [B, p1] without affecting the application.

Single Host Bus Adapters (HBA) to a multi-port drive

Consider a single path from the HBA through the fabric to a multi-port device.

Figure 3. Single HBA and multi-port drive



This configuration also provides a failover path unless the failure is with the HBA or the HBA's cable. At which point, the connection is severed and a permanent path error occurs. Whereas, if the failure occurs on the device side, an alternative path is still available for the information to go through which DPF automatically failovers to.

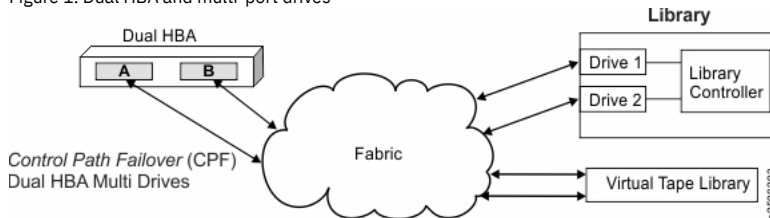
Control path failover

The following flowcharts outline the different types of configurations for control path failover (CPF). These configurations are presented in order of best practices as recommended by IBM.

Dual Host Bus Adapters (HBAs) to multi-port drives

Consider a multipath connection that consists of two Host Bus Adapters (HBAs) connected through a fabric to the library by at least two drives.

Figure 1. Dual HBA and multi-port drives



As seen in [Figure 1](#), four available paths are available between the drive and the host system. These paths are

- HBA A to drive 1 [A, d1]
- HBA A to drive 2 [A, d2]
- HBA B to drive 1 [B, d1]
- HBA B to drive 2 [B, d2]

As with DPF, one path is the primary path and the other three are alternate paths. If [A, d1] is the primary path and if HBA A fails, two remaining valid paths ([B, d1] and [B, d2]) are still available. The CPF attempts to switch to one of the available configured paths. Conversely, if the cable to drive 1 or drive 1 fails with [A, d1] as the primary path, two valid paths to the drive ([A, d2] and [B, d2]) are available. Without CPF support, if a permanent path error occurs (because of HBA or cable failover, for example), the application fails. With CPF, if a permanent failure with this configuration occurs, two valid physical paths are available for the data to be transmitted. Also, the running application is not affected.

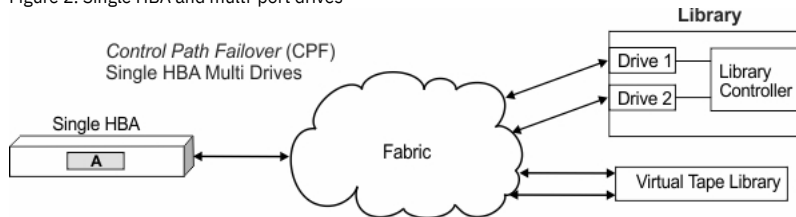
If the failed path is restored, the device driver claims the path as available and uses it as a valid alternate path in most conditions. This action is dependent on Operating System and HBA behavior, not the IBM® tape device driver behavior.

Note: In the operating systems logs, reservation conflict information might appear, which is because of scsi2 reservations that are not cleared. However, the device driver continues to try any paths that are available to make the reservation conflict transparent to the operating system.

Single Host Bus Adapter (HBA) to multi-port drives

Consider a single path from the HBA through the fabric to two drives in a library.

Figure 2. Single HBA and multi-port drives



This configuration also provides a failover path unless the failure is with the HBA or the HBA's cable. At which point, the connection is severed and a permanent path error occurs. Whereas, if the failure occurs with a drive or a drive's cable, an alternative path is still available for the information to go through which CPF automatically failovers to.

Dynamic load balancing

The dynamic load balancing support optimizes resources for tape devices that have physical connections to multiple host bus adapters (HBA) in the same machine. When an application opens a device that has multiple configured HBA paths, the device driver determines which path has the HBA with the lowest usage. Then, it 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. Then, that path is assigned to the second application. The device driver updates the usage on the HBA assigned to the application when the device is closed. Dynamic load balancing uses 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 assigns the application to use HBA1. When a second application opens a tape drive for use, the device driver assigns the second application to use HBA2. A third application is assigned to HBA1 and a fourth application is assigned to HBA2. Two applications are assigned to HBA1 and two applications are assigned to HBA2.

If the first application finishes and closes the device, there is now one application with HBA1 and two applications with HBA2. When the next application opens a tape drive, it is assigned to HBA1, so again there are two applications with HBA1 and two applications with HBA2. Likewise, if the second application finishes and closes the device, HBA2 has one application that is assigned to it. The next application that opens a tape drive is assigned to HBA2.

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

Dynamic Runtime Attributes

There are frequently field issues where customers must know which Initiator is holding a reservation in a drive or preventing the media from being unloaded. Also, they must correlate which drive special file name is used for the drive (such as rmt2). Sometimes this issue occurs over transport bridges and translators, losing any transport identifiers to help in this effort. LTO5, 3592 E07 (Jag 4) and later physical tape drives support attributes that are set to the drive dynamically by a host. This function is called Dynamic Runtime Attributes (DRA).

This feature is enabled by default. The attributes are set in the drive by the host during open, close, device reset, and data path change only. If there is a problem with sending the attributes to the drive, the error is ignored and not returned to the application.

There is no *ioctl* in the IBM® tape drivers to retrieve the dynamic runtime attributes but is an upcoming command on ITDT. The attributes can also be retrieved through a pass through *ioctl* to issue Read Dynamic Runtime Attributes SCSI command (see applicable *IBM Tape Drive SCSI Reference*). See the host platform section for any special information that pertains to the driver that concerns DRA. If there is a question whether your driver level supports DRA, see the fixlist that comes with your driver to see whether it was added. Updates are also required with the drive firmware.

Data encryption

- [Tape and library requirements](#)
- [Planning for application-managed tape encryption](#)
- [Planning for library-managed tape encryption](#)
- [Encryption feature codes](#)

Tape and library requirements

For specific encryption support, refer to SSIC at <http://www.ibm.com/systems/support/storage/config/ssic/>.

The following three major elements comprise the tape drive encryption solution.

- The encryption-enabled tape drive
The 3592 Model E07 and newer model tape drives, and the LTO Ultrium 4 and newer Ultrium drives are encryption capable. To run hardware encryption, the tape drives must be encryption-enabled. Encryption can be enabled on the encryption-capable tape drives through the Tape Library Specialist Web interface. Refer to the appropriate section in the documentation for your library for information about how to enable encryption.
Note: FC 1604, Transparent LTO Encryption, is required to use library-managed encryption on LTO Ultrium 4 and newer tape drives. It is not required for application-managed encryption. Refer to the sections on each method of encryption for information.
- Encryption key management
Encryption involves the use of several kinds of keys, in successive layers. How these keys are generated, maintained, controlled, and transmitted depends upon the operating environment where the encrypting tape drive is installed. Some data management applications, such as Tivoli® Storage Manager, can run key management. For environments without such applications or where application-agnostic encryption is wanted, IBM provides a key manager (such as the Tivoli Key Lifecycle Manager or the IBM® Security Key Lifecycle Manager) to complete all necessary key management tasks.
- Encryption policy
The method that is used to implement encryption. It includes the rules that govern which volumes are encrypted and the mechanism for key selection. How and where these rules are set up depends on the operating environment.

The LTO Ultrium 6 and later encryption environment is complex and requires knowledge beyond that of product trained Service Support Representatives (SSRs). The Encryption function on tape drives (desktop, stand-alone, and within libraries) is configured and managed by the customer. In some instances, SSRs are required to enable encryption at a hardware level when service access or service password controlled access is required. Customer setup support is by Field Technical Sales Support (FTSS), customer documentation, and software support for encryption software problems. Customer 'how to' support is also provided with a support line contract.

In the open system environment, there are two methods of encryption management to choose from. These methods differ in where you choose to locate your encryption key manager application. Your operating environment determines which is the best for you, with the result that key management and the encryption policy engine might be in any one of the three environmental layers: application layer, system layer, and library layer.

Application-managed tape encryption

This method is best where operating environments run an application already capable of generating and managing encryption policies and keys, such as Tivoli Storage Manager (TSM). Policies specifying when encryption is to be used are defined through the application interface. The policies and keys pass through the data path between the application layer and the Encryption is the result of interaction between the application and the encryption-enabled tape drive, and is transparent to the system and library layers.

Refer to [Planning for application-managed tape encryption](#) for details on the hardware and software requirements for application-managed encryption. For details on setting up application-managed tape encryption refer to the Tivoli Storage Manager documentation or for information, visit <http://publib.boulder.ibm.com/infocenter/tivihelp/v1r1/index.jsp>.

It is required to use the latest device drivers available. Refer to [Accessing documentation and software online](#) for downloading drivers. In different operating system environments, refer to the applicable chapter for each operating system.

Library-managed tape encryption

This method is best for encryption-capable tape drives in open attached IBM tape libraries. Scratch encryption policies that specify when to use encryption are set up through the IBM Tape Library Specialist Web interface. Policies are based on cartridge volume serial numbers. Key generation and management are run by an encryption key manager. Policy control and keys pass through the library-to-drive interface, therefore encryption is transparent to the applications.

Refer to [Planning for library-managed tape encryption](#) for details on the hardware and software requirements for library-managed encryption. For details on setting up library-managed encryption on encryption-capable tape drives, refer to the *IBM Tape Library Operator's Guide* for your library.

Planning for application-managed tape encryption

Note: Contact your IBM® representative for information about encryption on the IBM encryption-capable tape drive. To run encryption on the encryption-capable tape drive, the following is required.

- Encryption-capable tape drive
- Encryption configuration features:
 - Library code updates and Transparent LTO Encryption feature code for encryption-capable libraries
 - Tape drive code updates

Application-managed tape encryption setup tasks

Any task that is not identified as an IBM service task is the responsibility of the customer.

1. Install, cable, and configure the encryption-capable tape drive (refer your *IBM Tape Drive or Library Operator's Guide*)
2. Install appropriate IBM tape device driver level (Atape, for example).
3. Set up encryption policies. Refer to the appropriate Tivoli® Storage Manager documentation.
4. Perform write/read operation to test encryption.
5. Verify encryption of the test volume by Autonomic Management Engine (AME): issue **QUERY VOLUME FORMAT=DETAILED**

Verify that Drive Encryption Key Manager is set to Tivoli Storage Manager.

Planning for library-managed tape encryption

Note: Contact your IBM® representative for information about encryption on the IBM encryption-capable tape drive.

To complete encryption on the encryption-capable tape drive, the following items are required.

- Encryption-capable tape drive
- Keystore (Refer to documentation on Security Key Lifecycle Manager (SKLM))
- Encryption configuration features
 - Security Key Lifecycle Manager (SKLM)
 - Tape system library code updates and Transparent LTO Encryption feature code for encryption-capable libraries
 - Tape drive code updates

Library-managed tape encryption tasks

Any task that is not identified as an IBM service task is the responsibility of the customer.

1. Install, verify, and configure
 - a. Keystore
 - b. EKM (Refer to documentation on Security Key Lifecycle Manager (SKLM)) for information on both.
2. Install and cable the encryption-capable tape drive (IBM service task for TS1120 Tape Drive).
3. Use IBM tape library specialist to enable the tape drive for library-managed tape encryption (refer to your *IBM Tape Drive* or *Library Operator's Guide*).
4. Use library diagnostic functions to verify.

Bulk rekey

For customers with Library-Managed Encryption with 3592 Enterprise tape drives and IBM tape and changer drivers that are running on open systems operating system (AIX®, HP-UX, Linux®, Solaris, Windows), sample code for completing bulk rekey operations is available. The sample code packages are provided "as-is" with limited testing, and are provided to give customers guidance on bulk rekey operations.

For UNIX operating systems, a sample script (rekey_unix.sh) is provided and must be used with the *tapeutil* version that is bundled in the same package. For Windows operating systems, a sample c program (rekey_win.c) is provided. Both of these sample programs must be used with both the IBM tape and changer drivers. In addition, data cartridges must be in storage cells, not in I/O station cells or tape drives.

For information and to download the sample code packages, see <http://www.ibm.com/support/fixcentral/>.

Encryption feature codes

To use system-managed and library-managed encryption, the Transparent LTO Encryption feature codes that are listed in [Table 1](#) are required for the associated IBM® tape libraries with encryption-capable tape drives. If the drives in use are TS1120 tape drives, this feature code is not required for system-managed or library-managed encryption. If you are using application-managed encryption, no feature code is required on any encryption-capable tape drives.

Table 1. Feature codes
(encryption)

Tape library	Feature code
TS4500	FC 1604
TS3500	FC 1604
TS4300	FC 5900

Recommended access order (RAO) open function

RAO was first introduced on IBM enterprise tape drives in the 3592-E07 (TS1140). RAO enables tape control applications to accelerate the retrieval of a certain number of files from a single tape thereby reducing the seek time between those files.

A feature of the LTO-9 and later full-height drives is the ability to accept a list of User Data Segments and reorder those User Data Segments into a recommended access order that minimizes the locate portion of the time to read those User Data Segments. This sorted list is called a Recommended Access Order (RAO) list. A User Data Segment (UDS) is defined as a grouping of contiguous logical objects (i.e., logical blocks and filemarks) and is described by partition number, beginning logical object identifier, and ending logical object identifier.

The RAO implementation in LTO produces the best results for performance enhancement when there is little variability in block size or data compression ratio. When the variability in compression ratio or block sizes increase, the accuracy of the locate estimates may be reduced and any potential performance enhancements may be diminished.

AIX Tape and Medium Changer device driver

This chapter describes the IBM® AIX® Enhanced Tape and Medium Changer Device Driver (Atape) for IBM tape devices.

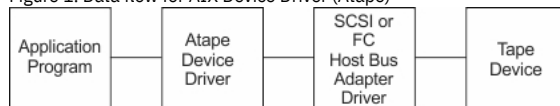
Purpose

The IBM AIX Enhanced Tape and Medium Changer device driver is designed to take advantage of the features that are 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 unmount the cartridges), and 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 that is described in this chapter covers the AIX Enhanced Device Driver (Atape device driver) and the interface between the application and the tape device. For data flow, refer to [Figure 1](#).

Figure 1. Data flow for AIX Device Driver (Atape)



- [Product requirements](#)
- [Installation and configuration instructions](#)
- [Tape drive, media, and device driver parameters](#)
- [Special files](#)
- [Persistent Naming Support](#)
- [Control Path failover support for tape libraries](#)
- [Data Path failover and load balancing support for tape drives](#)
- [System-managed encryption](#)
- [Problem determination](#)
- [Tape drive service aids](#)
- [Performance considerations](#)

Product requirements

Hardware requirements

Refer to [Hardware requirements](#) the latest hardware that is supported by the Atape device driver.

Software requirements

The AIX® Enhanced device driver (Atape device driver) supports AIX 5L Version 5.3 and later releases on IBM® POWER-based AIX servers.

For current software requirements, refer to the [Software requirements](#).

Installation and configuration instructions

The recommended procedure for installing a new version of the device driver is to uninstall the previous version.

Instructions for uninstalling the device driver are outlined in [Uninstalling](#).

1. At the end of the installation procedure, the installp facility automatically runs the AIX® **bosboot** command to update the boot record with the newly installed Atape files. When the **bosboot** command completes, the following messages are displayed:

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

This message refers 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 currently.

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

2. During the Atape install, the following entries are entered into the two system files.
 - /usr/lpp/bosinst/cdfs.optional.list to help the system image backup to DVD/CD mediaThe entry list in /usr/lpp/bosinst/cdfs.optional.list:

```
/usr/lib/drivers/Atape /usr/lib/drivers/Atape Atape.driver
/usr/lib/methods/cfgAtape /usr/lib/methods/cfgAtape Atape.driver
/usr/lib/methods/ucfgAtape /usr/lib/methods/ucfgAtape Atape.driver
/usr/lib/methods/defAtape /usr/lib/methods/defAtape Atape.driver
/usr/lib/methods/udefAtape /usr/lib/methods/udefAtape Atape.driver
/usr/lib/methods/chgAtape /usr/lib/methods/chgAtape Atape.driver
```

- /usr/lpp/bosinst/tape/tapefiles1 to create a bootable tape
- The entry list in /usr/lpp/bosinst/tape/tapefiles1:

```
/usr/lib/drivers/Atape
/usr/lib/methods/ucfgAtape
/usr/lib/methods/cfgAtape
/usr/lib/methods/udefAtape
/usr/lib/methods/defAtape
/usr/lib/methods/chgAtape
```

The entries are removed from the files when Atape is uninstalled.

Attention: The entries might be lost when a user upgrades the AIX file set of bos.sysmgt.sysbr for System Backup and BOS Install Utilities after Atape installation. It is recommended that the user check whether the entries still exist and add the entries into the files if needed.

- [Installation procedure](#)

- [Configuring Tape and Medium Changer devices](#)
- [Configuring limitations](#)
- [Deconfiguring tape devices](#)
- [Deconfiguring Medium Changer devices](#)
- [Uninstalling](#)

Installation procedure

For information on obtaining the latest version of device drivers and the latest documentation, refer to [Accessing documentation and software online](#).

Preinstallation considerations

Before the installation starts, verify the following items:

1. The tape device is properly functioning, properly attached to the server, and is powered up.
2. You logged on to the server on an account that has *root* authority.
3. You have a command shell window open on the server to run the installation procedure.
4. Make sure that the current path is defined in the command shell PATH variable. This definition can be accomplished in the Korn shell by 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. The following command is an example: **rmdev -l ost1 -d**

Installation procedure

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

```
lsllpp -l Atape.driver
```

Enter the following command to install the Atape driver in the current directory. For example

```
installp -acXd Atape.x.x.x.x Atape.driver
```

This command installs and commits the Atape driver on the system.

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 the following steps are completed.

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

Configure a tape device by 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).

- 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.

Configuring limitations

The subsequent limitations are applied for the Atape driver that runs on an AIX® host.

Maximum supported number of tape devices	1024
Maximum supported number of HBA ports	32
Maximum supported number of paths for a device (DPF/ CPF)	16/16
Maximum LUN size per target for FC HBA*	4095

Note: *On AIX systems, the maximum LUN number is 4095. Since Atape supports up to 1024 devices, Atape configures a total of 1024 devices by using the range from LUN 0 - 4095. For instance, a device with LUN 4095 at a SCSI target address can be configured by Atape if the total number of devices on the system is less than 1024.

Every opened tape device uses a certain amount of resources. The user must also consider other resources such as physical memory and virtual space on the system before you attempt to reach the limits.

Deconfiguring tape devices

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

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

1. The first method leaves the tape device that is defined in the configuration database. It is similar to bringing the device offline (not in use). Enter the following command to bring the /dev/rmt n tape device offline, but leave it defined in the device database.

```
rmdev -l rmt $n$ 
```

2. The second method brings the tape device offline and removes its definition from the device database. Enter the following command.

```
rmdev -l rmt $n$  -d
```

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

Deconfiguring Medium Changer devices

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

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

1. The first method leaves the device that is defined in the configuration database. It is similar to bringing the device offline. Enter the following command to bring the /dev/smc n medium changer device offline, but leave it defined in the device database.

```
rmdev -l smc $n$ 
```

2. The second method brings the medium changer device offline and removes its definition from the device database. Enter the following command.

```
rmdev -l smc $n$  -d
```

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

Uninstalling

Attention: All devices that use the Atape driver must be closed and cannot be in use when Atape is uninstalled or the uninstall fails.

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

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

Tape drive, media, and device driver parameters

This chapter describes the parameters that control the operating modes of the AIX® Enhanced Tape and Medium Changer Device Driver.

- [Configuration parameters](#)
- [Media parameters](#)

Configuration parameters

The operating parameters for the tape drive and device driver are set and changed by configuration parameters. The installation defaults are provided for all parameters initially. The AIX® **smit** command is used to set these parameters when a device is configured or to change these parameters. The AIX **chdev** command is 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 temporarily changed 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

- Alternate Pathing
- Autoloading
- Emulate autoloader (359x devices only)
- Block size
- Buffered mode (359x devices only)
- Compression
- Fail degraded media (359x devices only)
- Logical write protect (359x devices only)
- Logging
- Maximum size of the log file
- New logical name
- Read error recovery time (359x devices only)
- Record space mode
- Reservation key
- Reservation support
- Reservation type
- Retain reservation
- Rewind immediate

- System encryption
- System encryption for Write Commands
- Trailer labels
- SCSI status busy retry
- iostat support for tape

Alternate pathing

This parameter enables or disables the path failover support when a device is configured. See [Data Path failover and load balancing support for tape drives](#) for a description of the path failover and failover support.

The installation default is no (path failover is not enabled).

Autoloading

This parameter enables the autoloading feature of the device driver. It is used with the autoloading capability of the autoloader, ACF, ACL, or CSL installed on the tape device.

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.

Note: The autoloading feature is supported only on the following device types and configurations:

- IBM 3490E Models C11, C22, E01, E11, F01, and F11
- IBM Enterprise Tape System 3590, Models B11, E11, and H11
- IBM Magstar® MP 3570 Models B01, C01, B11, and C11
- IBM Magstar MP 3570 Models B02, B12, C02, and C12 (configured in split mode only)
- IBM 7332 (all models)

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 device is installed in a 3494 Enterprise Tape Library.
- The device is used by an application with stack loader support.
- The application is MKSYSB.
- The tapes that are read were not written with the autoloading feature.

Tapes that are created with AUTOLOAD=YES are not readable in configurations without Atape autoloader enabled, or on other UNIX operating systems, or on device types/models that are different from the backup device type/model.

If the parameter is set to On, then the tape stacker acts 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 driver automatically rewinds and unloads the tape, then loads the next tape. Then, it continues reading or writing the next tape. The following conditions are required to use this feature:

- The autoloading parameter must be set to On.
- The cartridge stacker must be loaded with one or more tapes.
- The ACF, ACL, or CSL must be set to Automatic, System, or Random mode.

This feature allows multivolume backups (with commands such as **tar**) without prompting for a volume change.

The installation default is Off (no autoloading).

Emulate autoloader

This parameter controls how the device driver operates when the ACF on the IBM Enterprise Tape System 3590, the IBM Magstar MP tape device, or the IBM 3490E Model Fxx is set to Random mode. If this parameter is set to On and the ACF is in Random mode, the device driver emulates an autoloading tape drive. When an unload command is sent to the device driver to unload a tape, the tape is unloaded, returned to the magazine, and the next tape in the magazine is loaded automatically into the tape drive. If this parameter is set to Off, the normal unload operation occurs, and the tape remains in the drive.

The emulate autoloader parameter can be used for legacy applications that are written for the IBM 3490E Automated Cartridge Loader (ACL) when the IBM Enterprise Tape System 3590, the IBM Magstar MP 3570, or the IBM 3490 Model F autoloader is set to Random mode. This parameter eliminates the need to reconfigure the autoloader of the device Random or Automatic operation.

The installation default is Off (do not emulate autoloader).

Note: On IBM Magstar MP 3570 Models B02, C02, and C12, this feature is supported only when the two drives are configured in Split mode, or in Base mode with one drive that is configured and available to AIX. This feature does not work in Base mode if both drives are in the available state to AIX.

Block size

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

The installation default is zero (use variable length) except for the IBM 7332 4-mm Tape Cartridge Autoloader, for which the default is a fixed block size of 1024 bytes.

Buffered mode

When a **write** command is processed, the data is either stored directly on the physical tape or buffered in the tape device. Buffering can increase the device performance.

The installation default is On (use Buffered mode).

Compression

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

The installation default is On (use compression).

Fail degraded media

This parameter controls whether the device driver fails a tape operation when degraded media is detected by the IBM Enterprise Tape System 3590. If a tape is loaded and the IBM 3590 cannot read the positioning information from the tape, the device driver is notified when the first command is sent to the tape drive. If this parameter is set to On, the device fails the command and returns a media error to the application. If this parameter is set to Off, the device driver does not fail the command.

Degraded media is a correctable condition that prevents the IBM Enterprise Tape System 3590 from running high speed Locate operations. A **Locate** command can take over 20 minutes, depending on the wanted position and the amount of data on the tape. This parameter is intended for use by real-time applications that cannot tolerate long **Locate** commands.

The installation default is Off (do not fail the tape operation if degraded media is detected).

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 information, refer to [Device and volume information logging](#).

The installation default is Off (no logging).

Maximum size of the log file

This parameter specifies the number of entries that are 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 stays constant. Each time a new entry is made, the oldest entry is overlaid. For information, refer to [Device and volume information logging](#).

The installation default is 500.

New logical name

Setting this parameter changes the logical name of the device to a new name as specified. After the logical name is changed, the new logical name parameter is cleared. For information, refer to [Persistent Naming Support](#).

There is no installation default value for this parameter.

Read error recovery time

This parameter controls the read error recovery time for the IBM Enterprise Tape System 3590. If this parameter is set to On, the recovery time for read errors is limited to a maximum of 5 seconds. If this parameter is set to Off, full recovery is used by the device and can take up to 10 minutes. This parameter is intended for use by real-time applications that cannot tolerate long delays when data is read from the tape.

The installation default is Off (do not limit the read error recovery time).

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.

Reservation key

This parameter specifies the SCSI Persistent Reservation key that is used by the device driver when either the Reservation Type parameter is SCSI Persistent Reserve and the Alternate Pathing parameter is set to no or when the Alternate Pathing parameter is set to Yes.

The default for this attribute is blank (NULL).

If the Reservation Key parameter is specified as blank (NULL), then the device driver uses an internal unique key for all devices on the host they are configured on. Another AIX host that shares devices also have an internal unique key for all devices if the Reservation Key parameter was blank (NULL).

If the default is not used, then the Reservation Key value can be specified as either a 1-8 character ASCII alphanumeric key or a 1-16 hexadecimal key that has the format 0xkey. If fewer than 8 characters are used for an ASCII key (such as host1), the remaining characters are set to 0x00 (NULL). If less than a 16 hexadecimal key is used, the remaining bytes are set to 0x00.

Note: When a Reservation Key is specified on each host that shares a device, the key must be unique to each host.

Reservation support

The parameter of `reserve_support` indicates that the Atape driver manages the reservation for the tape device when it is enabled. Atape reserves the tape device in open and releases it in close, and maintains the reservation in error recovery procedure (ERP).

Note: For the medium changer, this parameter is not applied when the Alternate Pathing (path failover) parameter is set to Yes. The device driver forces the setup to be disabled and the medium changer is not reserved in open when the Alternate Pathing parameter is set to Yes.
The installation default is Yes.

Reservation type

This parameter specifies the SCSI Reservation type that is used by the device driver, either a **SCSI Reserve 6** command or a **SCSI Persistent Reserve** command.

Note: This parameter is not used if the Alternate Pathing (path failover) parameter is set to Yes. The device driver uses SCSI Persistent Reserve when the Alternate Pathing parameter is set to Yes.

The installation default is SCSI Reserve 6.

Retain reservation

When this parameter is set to 1, the device driver does not release the device reservation when the device is closed for the current open. Any subsequent opens and closes until the **STIOCSETP** IOCTL is issued with `retain_reservation` parameter set to 0. The device driver still reserves the device on open to make sure that the previous reservation is still valid.

The installation default is Off (the reservation is released in close).

Rewind immediate

This parameter turns the immediate bit On and Off in rewind commands. If it is set to On, the rewind tape operation runs faster. However, 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) except for the IBM 7332 4-mm Tape Cartridge Autoloader, for which the default is On (rewind immediate).

System encryption

This parameter specifies whether System-Managed Encryption must be used. For information, refer to [System-managed encryption](#).

The installation default is No.

System encryption for Write commands

This parameter controls if System-Managed Encryption is used for Write commands. For information, refer to [System-managed encryption](#).

The installation default is Custom.

Trailer labels

If this parameter is set to On, then writing a record past the early warning mark on the tape is allowed. The first write operation to detect 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 by 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).

SCSI status busy retry

Atape retries the SCSI command fail due to the SCSI status Busy when the parameter of `busy_retry` is set to On. Otherwise, Atape fails the SCSI command if it is set to Off.

The installation default is Off.

iostat support for tape

The **iostat** command is used to monitor system input/output (I/O) devices. To work this system command for tape, Atape reports input and output statistics on each tape drive.

The installation default is On.

Media parameters

The ability to set or change media parameters is a tape diagnostic and utility function, refer to [IBM Tape Diagnostic Tool \(ITDT\)](#).

The media parameters can be queried and set by ITDT or the tape diagnostic and utility function by using the Query/Set Parameters option in the window.

These parameters cannot be set or changed by the configuration procedures. The media parameters are

- Capacity scaling

- Logical write protect
- Volume ID for logging
- Archive mode unthread (AMU)

Capacity scaling

This parameter sets the capacity or logical length of the current tape on IBM® Enterprise Tape System 3590, IBM Enterprise Tape System 3592, or Magstar® MP tape subsystems. By reducing the capacity of the tape, the tape drive can access data faster at the expense of data capacity.

Capacity scaling can be set at 100% for the entire tape (which is the default) or set at 75%, 50%, or 25% of the tape or any device-specific hexadecimal value. For example, on IBM 3592, to change capacity scaling from a 300 GB format tape (100%) to a 60 GB format tape, select the capacity scaling option. Then, select the option to enter a hexadecimal value and enter 35. Capacity scaling remains with the tape across mounts until it is changed.

Note:

1. The tape position must be at the start of the tape to change this parameter from its current value.
2. Changing this parameter deletes any existing data on the tape.
3. Attempting to set capacity scaling that is not supported by a device or the current loaded media always returns 100% and cannot be changed. For example, 60 GB media for the IBM 3592 cannot be capacity scaled and is always 100%.

Logical write protect

This parameter sets or resets the logical write protect of the current tape on IBM Enterprise Tape System 3590, IBM Enterprise Tape System 3592, or Magstar MP tape subsystems. The three types of logical write-protect are associated-protect, persistent-protect, and write-once read-many (WORM)-protect.

Associated protect remains only while the current tape is mounted or associated with the tape drive. It is reset when the tape is unloaded or the tape drive is reset.

Persistent protect remains or persists with the tape across mounts until it is reset.

WORM protect also remains with the tape across mounts, but (unlike persistent protect) it cannot be reset on the tape. After a tape is WORM protected, it can never be written on again.

Note:

1. The tape position must be at the start of the tape to change this parameter from its current value.
2. Attempting to set logical write protect that is not supported by a device or the current loaded media always returns "No" and cannot be changed.

Volume ID for logging

This parameter is the volume ID of the current 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.

Archive mode unthread (AMU)

This parameter turns Archive mode unthread (AMU) On and Off. When it is set to On, Atape manages and works the AMU feather that rewinds the tape cartridge to the end of tape at a low tension for long-term storage. The AMU feature on the tape drive is enabled for 3592 tape drives and disabled for LTO tape drives by default. To enable or disable this feature on the tape drive, the parameter must be turned on in the Atape driver.

Special files

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

- [Special files for tape devices](#)
- [Special files for Medium Changer devices](#)

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 1](#), in addition to the tape special files, a special file is provided for tape devices that allow access to the medium changer as a separate device.

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

For tape drives with attached SCSI medium changer devices, the rmt*.smc special file provides a separate path for commands that are issued to the medium changer. When this special file is opened, the application can view the medium changer as a separate SCSI 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 that are designed for a tape drive

If a tape drive has a SCSI medium changer device that is attached, then all operations (including the medium changer operations) are supported through the interface to the rmt* special file. For detailed information, refer to [Table 1](#).

Table 1. Special files for tape devices

Special file name	Rewind on Close (Note 1)	Retension on Open (Note 2)	Bytes per Inch (Note 3)	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 (Note 4)	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 (Note 5)	Yes	No	N/A	No	No
/dev/rmt*.smc (Note 6)	N/A	N/A	N/A	N/A	N/A

Note:

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

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 1](#) shows the attributes of the special file. The asterisk (*) represents a number that is 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. For information, refer to [Table 1](#).

Table 1. Special files for Medium Changer devices

Special file name	Description
/dev/smc*	Access to the medium changer robotic device
/dev/smc*.null	Pseudo medium changer device

Note: The smc*.null file is a pseudo device similar to the /dev/null AIX® special file. The commands can be issued to this file without a real device that is attached to it, and the device driver returns a successful completion. This file can be used for application development or debugging problems.

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

- Read
- Write
- Commands that are designed for a tape device

Persistent Naming Support

Persistent naming support is used to ensure that attached devices are always configured with the same logical name based on the SCSI ID, LUN ID, and host bus adapter (HBA), even when the system is rebooted.

When the AIX® operating system is booted, the HBA runs a device discovery and assigns a default logical name to each device found in a sequential order. If there are three tape drives attached to a parallel SCSI adapter, each with a LUN ID of 0 and a target address of 0, 1, and 2, the HBA initially configures them as Available with the following logical names.

```
rmt0    target 0, lun 0    Available
rmt1    target 1, lun 0    Available
rmt2    target 2, lun 0    Available
```

Run the following commands before the machine is rebooted.

```
- rmdev -dl rmt1
- rmdev -dl rmt2
```

On the next reboot, if the existing rmt1 target 1 device is powered off or not connected, the HBA initially configures two devices as Available with the following logical names:

```
rmt0    target 0, lun 0    Available
rmt1    target 2, lun 0    Available
```

If the previous rmt1 target 1 device is powered on after reboot and the **cfgmgr** command is run, the HBA configures the device as rmt2 instead of rmt1.

```
rmt2    target 1, lun 0    Available
```

This is one example, but there are other cases where the logical names of devices could change when the system is rebooted. For applications that need a consistent naming convention for all attached devices, it is accomplished with persistent naming support by defining a unique logical name (other than the AIX default names) that are associated with the specific SCSI ID, LUN ID, and HBA that the device is connected to.

- [Changing the logical name after initial boot](#)

Changing the logical name after initial boot

The logical name of a device can be changed after an initial boot and configured. This procedure can be done by using the SMIT menu or the **chdev** command from a script or command line.

For example, a default rmt0 logical name for a tape drive can be changed to rmt-0, tape0, or any descriptive name wanted. In this example, if the three tape drives are changed to rmt-0, rmt-1, and rmt-2, and the system is then rebooted with rmt-1 powered off, the HBA detects that unique names are predefined for the attached devices, and the HBA uses those names. In this case, the devices configure as follows:

```
rmt-0    target 0, lun 0    Available
rmt-1    target 1, lun 0    Defined
rmt-2    target 2, lun 0    Available
```

Since rmt-1 is not detected by the HBA but is predefined at the SCSI ID and LUN ID, it remains in the defined state and is not configured for use. But, the next rmt-2 tape drive configures as the same name at the same location after reboot.

Changing the logical name with SMIT

To change the logical name by using SMIT, complete the following steps:

1. Run SMIT from a command line and select Devices.
2. Select Tape Drive.
3. Select Change/Show Characteristics of a Tape Drive.
4. Select the logical device to be changed from the list displayed.
5. In the New Logical Name field, enter a non-AIX default logical name.
6. Press Enter to process the change.

Changing the logical name with the chdev command

The logical name of a device can be changed by using the **chdev** command. For example, to change the logical name of the device from rmt0 to rmt-0, run

```
chdev -l rmt0 -a new_name=rmt-0
```

The output of this command displays

```
rmt0 changed
```

Note:

- When path failover is enabled, if you change the logical name for either a primary or alternate device, only the individual device name changes.
- Follow the naming convention whenever you run mksysb, bosboot:
 - The prefix name of "rmt" cannot be changed.
 - A sequence number must be a positive integer. The smallest sequence number is 0.
 - The prefix name cannot contain non-numerical characters. For example, rmt1_rescu is not an acceptable prefix name.
 - When a device instance logical name is generated, the SMIT tool automatically assigns the next available sequence number (relative to a specific prefix name). The next available sequence number is defined as the smallest sequence number for a particular prefix name that is not yet allocated.

Control Path failover support for tape libraries

Note: The library control path failover feature code must be installed before enabling the path failover support in the Atape device driver. Refer to [Automatic failover](#) for what feature codes might be required for your machine type.

The Atape device driver path failover support configures multiple physical control paths to the same logical library within the device driver. It also provides automatic failover to an alternate control path when a permanent error occurs on one path. This support is transparent to the running application.

- [Configuring and unconfiguring path failover support](#)
- [Primary and alternative paths](#)
- [Querying primary and alternative path configurations](#)
- [Configuring and unconfiguring primary and alternative devices](#)

Configuring and unconfiguring path failover support

Path failover support is not enabled automatically when the device driver is installed. It must be configured initially on each logical device after installation. When path failover support is enabled for a logical device, it remains set until the device is deleted or the support is unconfigured. The alternate path failover 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, then select Yes or No for Enable Path Failover Support. The support can also be enabled or disabled by 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 alternative paths

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

When a second logical device is configured with path failover support enabled for the same physical device, it configures as an alternative path. On SCSI attached devices, -A is appended to the location field. On Fibre attached devices, -ALT is appended to the location field of the device.

A third logical device is also configured as an alternative path with either -A or -ALT appended, and so on. The device driver supports up to 16 physical paths for a single device.

The labeling of a logical device as either a primary or alternative path is for information only, to

1. Be able to identify the actual number of physical devices that are configured on the system and a specific logical device that is 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 an alternative path for the same devices.
2. Provide information about which logical devices configured on the system have path failover support enabled.

Querying primary and alternative path configurations

You can see the primary and alternative path configuration for all devices with the **lsdev** command. Two or more logical devices can be configured for a single physical device, but the first device that is configured is labeled the primary device. All other logical devices that are configured after the first device are labeled as alternative devices. To see this information, run the **lsdev -Cc tape | grep P** command and look at the location field in the data. Run the following command,

```
lsdev -Cc tape | grep P
```

For example, you can easily determine how many physical devices are configured with path failover support.

Note: Show the primary and alternative path configuration for any device by using tape diagnostic and utility functions. Refer to [IBM Tape Diagnostic Tool \(ITDT\)](#).

Configuring and unconfiguring primary and alternative devices

Logical devices that are configured as alternative paths can be unconfigured and reconfigured at any time after the initial configuration is run. Unconfiguring an alternative 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 alternative devices are still available.

Likewise, configuring a new alternative 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 alternative 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 alternative devices that were unconfigured are reconfigured. The first device that is reconfigured becomes the new primary device. All remaining alternative devices are reconfigured as alternative paths.

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

Data Path failover and load balancing support for tape drives

Note:

1. Some devices require a path failover feature code that is installed before the path failover support is enabled in the Atape device driver. Refer to [Automatic failover](#) for what feature code might be required for your machine type.
2. DPF keys do not need to be added if you are running the latest drive code on Ultrium 3 and Ultrium 4 drives.
3. This function is not supported for devices that are attached through an IBM® San Data Gateway or on the IBM Virtualization Engine TS7510.
4. The AIX® operating system supports only a static configuration of devices, which also applies to the Path Failover 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.

- [Installing Data Path failover license key](#)
- [Configuring and unconfiguring path failover support](#)
- [Primary and alternative paths](#)
- [Querying primary and alternative path configuration](#)
- [Configuring and unconfiguring primary and alternative devices](#)

Installing Data Path failover license key

Use the following command line script to query, add, or delete license keys for this feature before the path failover feature is enabled as described below. The key is a 16-digit hexadecimal value, for example, **1234567890abcdef**.

All key values “A-F” must 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 path failover support

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

Path failover 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 action unconfigures all devices that have path failover set to No, and reconfigures all devices, setting path failover 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 Path Failover Support. The support can also be enabled or disabled by using the **chdev** command, for example:

```
chdev -l rmt0 -aaltpathing=yes
```

```
chdev -l rmt0 -aaltpathing=no
```

Primary and alternative paths

When the device driver configures a logical device with path failover support enabled, the first device that is 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 path failover support enabled for the same physical device, it configures as an alternative path and ALT is appended to the location field. A third logical device is configured as the next alternative 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 is similar to the following command.

```
rmt0 Available 20-60-01-PRI IBM 3590 Tape Drive and Medium Changer (FCP)
rmt1 Available 30-68-01-ALT IBM 3590 Tape Drive and Medium Changer (FCP)
```

If rmt1 is configured first, then rmt0, the command output is similar to the following.

```
rmt0 Available 20-60-01-ALT IBM 3590 Tape Drive and Medium Changer (FCP)
rmt1 Available 30-68-01-PRI IBM 3590 Tape Drive and Medium Changer (FCP)
```

The labeling of a logical device as either a primary or alternative path is for information only, to

1. Identify the actual number of physical devices that are configured on the system and a specific logical device that is associated with them. Only one logical device is labeled the primary path for each physical device. However, many (multiple) logical devices can be labeled as an alternative path for the same devices.
2. Provide information about which logical devices configured on the system have path failover support enabled.

Querying primary and alternative path configuration

You can see the primary and alternative path configuration for all devices with the **lsdev** command. Two or more logical devices might be configured for a single physical device, but the first device that is configured is labeled the primary device. All other logical devices that are configured after the first device are labeled as alternative devices.

To see this information, 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 System p (also known as pSeries) server are configured with path failover support.

Configuring and unconfiguring primary and alternative devices

Logical devices that are configured as alternative paths can be unconfigured and reconfigured at any time after the initial configuration is run. Unconfiguring an alternative 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 alternative devices are still available. Likewise, configuring a new alternative 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 alternative 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 alternative devices that were unconfigured are reconfigured. The first device that is reconfigured becomes the new primary device. All remaining alternative devices are reconfigured as alternative paths.

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

System-managed encryption

- [Device driver configuration](#)
- [Querying tape drive configuration](#)
- [Testing data encryption configuration and connectivity](#)
- [Error logging](#)
- [Field support information](#)

Device driver configuration

System-managed encryption can be set on a specific tape drive by using the standard SMIT panels to Change/Show Characteristics of a tape device or the command line **chdev** command. There are two new attributes added for encryption.

sys_encryption "yes/no"	Use System Encryption FCP Proxy Manager
wrt_encryption "off/on/custom"	System Encryption for Write Commands at BOP

The **sys_encryption** attribute enables device driver system-managed encryption for a tape drive by setting the value to yes.

The **wrt_encryption** attribute controls whether the device driver can set the tape drive to encryption enabled for write commands. When set to off, the tape drive uses encryption for read operations; write operations do not use encryption. When set to on, the tape drive uses encryption for both read/write operations. When set to custom, the device driver does not modify current tape drive setting. The custom setting is intended for applications that use system-managed encryption to control write encryption without device driver intervention.

Note: If **wrt_encryption** is set to on, an application cannot open a tape drive by using the append mode.

Querying tape drive configuration

Querying the tape drive configuration is a tape diagnostic and utility function. Refer to [IBM Tape Diagnostic Tool \(ITDT\)](#).

Testing data encryption configuration and connectivity

A data encryption test is available to validate the ibmekm.conf file server entries and test tape drive to server connectivity operations.

This test is a tape diagnostic and utility function. Refer to [IBM Tape Diagnostic Tool \(ITDT\)](#).

Error logging

Encryption errors are logged along with other tape operation errors by using the standard TAPE_ERR1 Template "Tape Operation Error" with associated sense data in the detail data.

An encryption failure is indicated when the asc/ascq in the sense data is EFxx or EExx. Refer to the tape drive hardware reference for information on the asc/ascq being reported. The asc/ascq can be found in the first column of the second row in detail sense data. For example,

```
Detail Data
SENSE DATA
0A00 0000 5A08 25FF 0000 00FF FF00 0000 0000 0000 F000 0600 0000 1458 0000 0000
EF11 FF02 D105 0000 0009 0191 0002 0000 0000 0A00 0000 0000 0000 0000 0000
0000 0000 0000 FFFF FF00 0000 FFF0 B7E3 0001 2127 0000 0000 0000 0000 3930 3220
2020 2000 0041 4A00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Field support information

When encryption failures require field support or development analysis, the following data must be provided for a problem on a specific tape drive from the machine (rmt1 for example) for the device driver. Tape drive dumps and EKM server logs might be needed in addition to this information.

```
errpt -a > errpt.rmt1
lsattr -El rmt1 > lsattr.rmt1
All Atape files in /var/adm/ras/Atape*
```

Problem determination

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

- [Dump support](#)
- [Device and volume information logging](#)
- [Log file](#)
- [Tape log utility](#)
- [Reservation conflict logging](#)
- [Error logging](#)
- [Error log templates](#)
- [Automatic dump facility](#)
- [Dynamic Runtime Attributes](#)
- [Trace facility](#)
- [Atape System Trace \(ATRC\) utility](#)
- [Component tracing](#)
- [Atape Component Trace \(ACTRC\) utility](#)

Dump support

Dump support is provided through the dump entry point in the driver. Refer to appropriate AIX® manuals for a description of how to use the dump devices and how to read the dump data.

Dump device 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 run 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 the **sysdumpstart -s** command is issued.

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 the **crash** command is used 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.

Device and volume information logging

An optional logging utility is provided to log the information about the device and the media. The 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

Refer to [Tape drive, media, and device driver parameters](#) 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 pages that the device supports. Page 0x00 is followed by all pages that are 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 shown as hexadecimal values in dump format.

The syntax for the tape log utility is

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

Note:

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 contents of the log file are displayed as standard output. To save the log in a file, use the AIX® redirection function. For example,

```
tapelog -l rmt0 > rmt0.log
```

Reservation conflict logging

When the device driver receives a reservation conflict during open or after the device is opened, it logs a reservation conflict in the AIX error log. Before it logs the error, the device driver issues a **Persistent Reserve In** command to determine whether a SCSI Persistent Reservation is active on the reserving host to get the reserving host initiator WWPN (worldwide port name) and reserve key. If successful, the device driver logs this information as follows,

```
Reserving host key xxxxxxxx WWPN xxxxxxxx
```

Where the first xxxxxxxx is the actual reserve key, and the second xxxxxxxx is the reserving host initiator WWPN.

After the reserving host WWPN is initially logged, subsequent reservation conflicts from the same reserving host WWPN are not logged. This action prevents multiple entries in the error log until either the reserving host WWPN is different from the one initially logged or the device driver reserved the device and another reservation conflict occurs.

If the **Persistent Reserve In** command fails, the detail data contains the following entry with the errno from the failing **Persistent Reserve In** command.

```
Unable to obtain reserving host information, errno x
```

The possible *errno* values are

- ENOMEM - Device driver cannot obtain memory to run the command
- EINVAL - Device has a Persistent Reservation but does not support the Persistent Reserve In service action
- EBUSY - Device failed the command with reservation conflict and has an SCSI-2 Reserve active
- EIO - Unknown I/O failure occurred

Error logging

The device driver provides logging to the AIX® system error log for various errors. The error log can be viewed for specific devices by using the Error Log Analysis utility that is provided with the tape drive service aids. Refer to [Error Log Analysis](#). The error log can also be viewed by using the **smit** or the **errpt** command.

Error log templates

The error log templates the device driver uses 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 that is associated with the type of error. The following items describe the error labels and detail data for the templates that are 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
- SIM_MIM_RECORD_3590
3590 Service/Media Information Message (Log Page X '31')
- TAPE_SIM_MIM_RECORD
Tape drive Service/Media Information Message (Log Page X '31')
- DEV_DUMP RETRIEVED
Device dump retrieved
- TAPE_DRIVE_CLEANSING
Tape drive needs cleaning
- RESERVE_CONFLICT
Reservation conflict

Detail data

Detail data is logged with the associated error that identifies the cause of the error. Detail data for the SIM_MIM_RECORD_3590 or TAPE_SIM_MIM_RECORD entries contain the raw data from Log Sense Page 31. Refer to the hardware reference manual for the format of this entry. All other error log entries use the following format for detail data:

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

aa Length of the command descriptor block (CDB).

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 - 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, then 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 to 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 byte 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 that SCSI bus was reset.

For FCP or SAS adapters, this byte 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 worldwide 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.

- OA TRANSPORT BUSY - The transport layer is busy.
- OB TRANSPORT DEAD - The transport layer is inoperative.

ss

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

Automatic dump facility

The device driver provides an automatic dump facility for devices that support reading a dump and indicating when a dump is available in device sense data. Whenever a check condition occurs and the sense data indicates that 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 five dumps for each device are stored in this directory as

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

Note: *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.

Dynamic Runtime Attributes

Field issues can occur where customers must know which initiator on a host is holding a reservation in a drive or preventing the media from being unloaded. Also, they must correlate which special file name is used for the drive (such as *rmt2*). Sometimes this issue occurs over transport bridges and translators, thus losing any transport identifiers to help this effort. LTO5, 3592 E07, 3592 E08, and later physical tape drives support attributes set to the drive dynamically by a host. This function is called Dynamic Runtime Attributes (DRA). Atape supports the feature, starting from 12.5.9.0.

The feature is enabled by default. The drive attributes on the host are set during the open, close, device reset, and data path change only. So, it does not have impact on the tape and system performance. The error is ignored and is not returned to application, when the information failed to send. However, a device attribute named as "host_attributes" is available to disable the DRA feature, when Atape runs with the virtual tape library for which the feature is not supported.

Run the **lsattr** command to display the attribute of *host_attributes* setup.

On LTO tape drive:

```
# lsattr -El rmt1
host_attributes yes      Host Dynamic Runtime Attribute (LTO-5 and later only) True
```

On 3592 tape drive:

```
# lsattr -El rmt4
host_attributes yes      Host Dynamic Runtime Attribute for 3592-E07 and later True
```

To modify the attribute of "host_attribute", run the **chdev** command.

```
# chdev -lrmt1 -a host_attributes=no
rmt1 changed
```

Trace facility

The AIX® trace facility is supported for the device driver. The trace event is identified with a *hookword*. The hookword that is used by the device driver is 326. The trace can be initiated 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 run.

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.

Atape System Trace (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.

Component tracing

Later releases of AIX® 5.3 and above support component tracing. Unlike system tracing that must be started and stopped, component tracing by default is on all the time and runs continually.

To determine whether component tracing is available, run the command **ctctrl -q** to display a list of supported components with their default settings. You must have root authority to run this command. Refer to the AIX **ctctrl** man page for a complete description of the **ctctrl** command and parameters.

To dump and format the current component trace to a file (for example, **actrc.out**) into the current directory, run the following commands.

```
ctctrl -D -c Atape
trcrpt -l Atape -o actrc.out
```

The Atape component trace can also be retrieved from a system dump. This action eliminates the need to start the Atape system trace before a system dump or to re-create an AIX system dump when a system trace is not running. The AIX system dump is normally stored in the **/var/adm/ras** directory as a **vmcore.x.BZ** file, where **x** is a dump number 1, 2, and so on.

To retrieve and format the Atape component trace from a dump file (for example, **vmcore.1.BZ**) to a file (for example, **actrc.dump**) into the current directory, run the following commands.

```
dmpuncompress /var/adm/ras/vmcore.1.BZ
trcdead -c /var/adm/ras/vmcore.1
trcrpt -l Atape -o actrc.dump
```

Atape Component Trace (ACTRC) utility

The **actrc** component trace utility is also installed with the device driver to dump and format the current Atape component trace. To dump and format the component trace, run the command **actrc**. The trace is formatted to an **actrc.out** file in the current directory.

Tape drive service aids

The service aids described in the following sections are accessible through the AIX® diagnostic subsystem by using the AIX **diag** command and in the Task Selection menu by selecting IBM Tape Drive Service Aids. Refer to [Tape drive service aids details](#).

Tape drive service aids are also available by using the [IBM Tape Diagnostic Tool \(ITDT\)](#).

- [Tape drive service aids details](#)

Tape drive service aids details

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

- Force Microcode Dump
- Read Dump
- Microcode Load
- Error Log Analysis
- Reset Drive
- Create an FMR Tape

Force Microcode Dump

This utility forces a dump operation on the tape drive. After the dump operation is completed, the dump data can be transferred from the tape drive by using the Read Dump utility.

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/xmt0**). 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 Esc and the number 7 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 IBMTape 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 Esc and the number 7 instead of F7.
5. If the current microcode on a tape drive is Federal Information Processing Standard (FIPS) code, then a window opens and displays the following message.

```
Warning: The drive is currently using FIPS code. Press Enter to
continue with downloading new drive code.
```

If you do not want to download the new code, press either F3 to cancel or F10 to exit without downloading new code. Otherwise, press Enter to continue with the download code procedure.

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.

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 IBMTape Drive Selection window, then press Enter.
The Reset Drive operation starts, and a window opens when the operation is completed.

Create an FMR Tape

This utility creates a field microcode replacement (FMR) cartridge tape by using the loaded functional microcode in the tape drive.

To access this utility -

1. Open the Service Aids window.
2. Select Create an FMR Tape from the IBM Tape Drive Service Aids window, then press Enter.
3. Select the device from the IBMTape Drive Selection window, then press Enter.
The Create an FMR Tape operation starts, and a window opens when the operation is completed.

Performance considerations

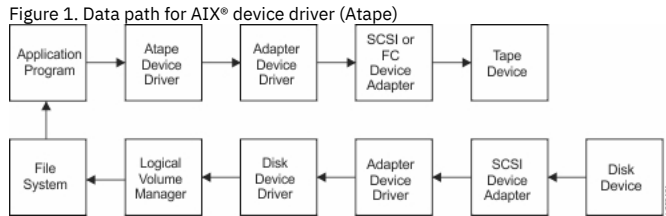
This chapter describes the parameters and issues that can affect the perceived performance of the tape drive. In general, AIX® applications that operate at a file level to move data between disk storage devices and tape do not use the full capabilities of a high end tape device. The goal of this discussion is to give an overview of the data path components that are 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 that are encountered by advanced application developers are beyond the scope of this document.

- Refer to the hardware reference for the specific device for performance specifications.
- Refer to the application documentation for information on device-specific application configuration.
- Refer to the operating system documentation for information on disk storage device striping and other techniques for improving file system performance.
- [Data path](#)
- [Common AIX utilities](#)
- [AIX iostat utility for tape performance](#)
- [Before Support is called](#)

Data path

The simplified model in [Figure 1](#) shows the components that are 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.



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 must 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 must be enabled for maximum performance if the data sent to the device is uncompressed.
2. The `block_size` parameter must be set to variable (`block_size=0`) and command or application parameters that are specified to a block size appropriate for the device.
3. Block sizes of 128 KB or greater must be used to improve performance.

AIX iostat utility for tape performance

In releases of AIX® 5.3 and earlier, the AIX `iostat` utility supports tape performance statistics in addition to other supported devices (such as disk). To determine whether the `iostat` utility supports the configured tape drives, run the command `iostat -p`. If the configured tape drives are supported, a list of configured tape drives are displayed with the statistics listed for each drive. Refer to the AIX `iostat` man page for a complete description of the `iostat` command and parameters. When the Data Path Failover feature is used, only the primary path for the tape drive is listed. The statistics apply to both the primary and alternative paths that are used.

Before Support is called

System performance tuning is not a support responsibility. If tests indicate that raw tape performance is below specifications, record the exact failing command. Then, collect the output from the commands in [Table 1](#) before support is contacted.

Table 1. Error description

Information	Command
Configuration	<code>lscfg -v</code>
Device parameters	<code>lsattr -E -l rmtN</code>
Error log. Call hardware support if errors are found for TAPE_ERR* or SCSI* error labels.	<code>errpt -a</code>
Driver version	<code>lspp -l Atape.driver</code>
Trace of failing command	Refer to Trace facility .

Linux Tape and Medium Changer device driver

This chapter describes the IBM® Linux® Tape and Medium Changer device driver (`lin_tape`).

For tape diagnostic and utility functions, refer to [IBM Tape Diagnostic Tool \(TDT\)](#).

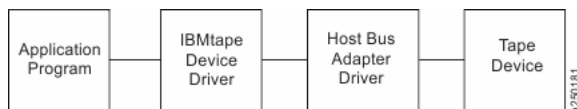
Purpose

The `lin_tape` and medium changer device driver is designed to take advantage of the features that are 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 unmount the cartridges), and also the advanced functions that are 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 that is described in this chapter covers the Linux device driver (`lin_tape` device driver) and the interface between the application and the tape device.

Figure 1. Data flow for Linux device driver (`lin_tape`)



- [Product requirements](#)
- [Installation and Configuration instructions](#)
- [Tape drive, media, and device driver parameters](#)
- [Special files](#)
- [Control Path failover support for tape libraries](#)
- [Data Path failover and load balancing support for tape drives](#)
- [Open source device driver - lin_tape](#)
- [System-managed encryption](#)
- [Problem determination](#)

Product requirements

Hardware requirements

For current hardware requirements, refer to the [Hardware requirements](#).

Software requirements

For current software requirements, refer to the [Software requirements](#).

Installation and Configuration instructions

The lin_tape device driver for Linux® is provided in a source rpm package. The utility tools for lin_tape are supplied in binary rpm packages. Refer to [Accessing documentation and software online](#). They are downloaded with the driver.

The following sections describe installation, configuration, uninstallation, and verification procedures for lin_tape and its utility tools. Refer to Linux documentation for **tar** command information and any Linux distribution that support rpm for **rpm** command information. You must have *root* authority to proceed with the installation of the driver. See the README file that can be downloaded with the driver at Fix Central. For information about downloading drivers, see [Accessing documentation and software online](#).

This file contains the latest driver information and supersedes the information in this publication.

- [Conventions used](#)
- [Configuration limitations](#)
- [Components created during installation](#)
- [Installation procedure](#)
- [Updating procedure](#)
- [Querying the installed package](#)
- [Configuring Tape and Medium Changer devices on Intel-compatible systems](#)
- [Configuring Tape and Medium Changer devices on IBM System p models](#)
- [Configuring Tape and Medium Changer devices on IBM System z models](#)
- [Uninstallation procedure](#)

Conventions used

In subsequent pages, you see file names with x.x.x in them. The x.x.x refers to the version of the driver, which changes as IBM® releases new driver levels. Use the actual driver version numbers as you complete the instructions.

Commands that you are to type are indicated with a leading ">" character, which indicates the shell prompt.

Note: This procedure is run with tape diagnostic and utility functions. See [IBM Tape Diagnostic Tool \(TTDT\)](#).

Configuration limitations

Maximum supported number of tape devices	1024
Maximum supported number of HBA ports	16 (8 dual-port, 4 quad-port)
Maximum supported number of paths for a device (DPF/CPF)	16/16
Maximum LUNs per system	256

Every attached tape or library device uses a certain amount of resources. The user must consider resources such as physical memory and virtual space on the system, which might further limit the number of devices that can be attached.

Components created during installation

The `lin_tape` package consists of the device driver and a number of associated files. Components that are created during `lin_tape` installation (from the rpm package) are listed in [Table 1](#).

Table 1. Linux: Components created during `lin_tape` installation

Component	Description
<code>/lib/modules/(Your system's kernel name)/kernel/drivers/scsi/lin_tape.ko</code> <code>/lib/modules/(Your system's kernel name)/kernel/drivers/scsi/pfo.ko</code> Only when <code>sfmp</code> flag is used during build:	Device driver module for current kernel version
<code>/lib/modules/(Your system's kernel name)/kernel/drivers/scsi/sgmp.ko</code> <code>/lib/modules/(Your system's kernel name)/kernel/drivers/scsi/stmp.ko</code>	
<code>/usr/bin/lin_taped</code>	<code>lin_taped</code> daemon
<code>/etc/lin_taped.conf</code>	<code>lin_taped</code> daemon configuration file
<code>/usr/share/doc/lin_tape/lin_tape_daemon.Readme</code> (for Red Hat)	Readme file for <code>lin_taped</code> daemon
<code>/usr/share/doc/packages/lin_tape/lin_tape_daemon.Readme</code> (for SUSE LINUX)	
<code>/usr/share/doc/lin_tape/lin_tape.Readme</code> (for RedHat)	Readme file for <code>lin_tape</code> driver
<code>/usr/share/doc/packages/lin_tape/lin_tape.Readme</code> (for SUSE LINUX)	
<code>/usr/share/doc/lin_tape/copying</code> (for Red Hat)	License documentation for <code>lin_tape</code>
<code>/usr/share/doc/packages/lin_tape/copying</code> (for SUSE LINUX)	

Note: On a Unified Extensible Firmware Interface secure boot enabled kernel, modules are required to be signed before they can be loaded. Refer to [Table 1](#) to identify the device driver modules and their location after installation. Proceed to sign the device driver modules by following the operating system instructions while keeping the modules at the same location.

Installation procedure

If `lin_tape` is already installed on your system, refer to the [Updating procedure](#). This section assumes that you are installing the `lin_tape` device driver onto a system where it is not currently installed.

If you are installing `lin_tape` on a system that is running Linux® for S/390® or Linux for zSeries, ensure that the OpenFCP adapter device driver `zfcp` is loaded in the kernel. Refer to [Configuring Tape and Medium Changer devices on IBM System z models](#) for how to configure and install `zfcp`.

Make sure that the C/C++ development and kernel development packages are installed on your system. To install the `lin_tape` driver with all the added value of the `lin_taped` daemon, complete the following steps.

1. Download the appropriate level of the source RPM package to a directory of your choice on the Linux kernel for which you want to install it.
2. Run `rpmbuild --rebuild <filename>`, where: `<filename>` is the name of the RPM file. A binary RPM package is created for your kernel from the source RPM package. For example,

```
>rpmbuild --rebuild lin_tape-1.x.x.x-1.src.rpm
```

Note: For the current `lin_tape` driver, it is possible to enable path failover for `st/sg` interfaces. See [Enabling path failover for st and sg interface](#) for details. If path failover is enabled and you want to also enable it for `st` or `sg` devices, an extra flag that is named `sfmp` must be added at this step:

```
>rpmbuild --rebuild -with sfmp lin_tape-1.x.x.x-1.src.rpm
```

3. Output from the build is printed to your screen. Near the end of the output, a line indicates the file name and location of your binary RPM package. For example, a line similar to the following is output to your screen.

```
Wrote: /usr/src/redhat/RPMS/i386/lin_tape-1.x.x.x-1.i386.rpm
```

4. To install the `lin_tape` driver from the binary package, run `>rpm -ivh <filename>` For example,

```
>rpm -ivh /usr/src/redhat/RPMS/i386/lin_tape-1.x.x.x-1.i386.rpm
```

Note: For Ubuntu, run `alien -i --scripts <filename>` (starting with `lin_tape` version 3.0.33).

5. To install the `lin_taped` daemon, download it to your Linux file system and run `rpm -ivh` on the daemon RPM file. For example,

```
>rpm -ivh /usr/src/redhat/RPMS/i386/lin_taped-1.x.x.x-1.rhel5.i386.rpm
```

Note: For Ubuntu, run `alien -i --scripts <lin_taped*.rpm>`, and then run `sudo update-rc.d lin_tape defaults`.

Updating procedure

If your current `lin_tape` device driver was installed from an rpm package previously, you can uninstall the driver first, then install the newer version. For example,

On SLES and RHEL:

```
>rpm -e lin_taped
```

```
>rpm -e lin_tape
```

Then, follow the installation procedure in the previous section.

On Ubuntu:

```
apt purge lin-taped
```

```
apt purge lin-tape
```

Then, follow the installation procedure in the previous section.

Note: All tape devices that use the lin_tape device driver must be closed and cannot be in use when lin_tape is uninstalled.

Querying the installed package

The query is supported for the lin_tape device driver rpm package and SLES and RHEL only.

The installed rpm package can be queried by running the following commands to show information that is associated with the package.

To show information about lin_tape -

```
>rpm -qi lin_tape
```

To show the file list, enter the command

```
>rpm -ql lin_tape
```

To show the states of files in the package, for example, *normal*, *not installed*, or *replaced* -

```
>rpm -qs lin_tape
```

To query lin_tape package on Ubuntu:

```
dpkg-query --list lin-tape
```

Configuring Tape and Medium Changer devices on Intel-compatible systems

Physically attach your tape and medium changer devices to your Linux® server.

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 a tape library with the integrated router, before the QLogic driver is installed, 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 does not recognize all of the attached devices. To view the LUNs of attached devices, log on to 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.

When you run the lin_tape kernel module, it creates special files in the /dev directory.

Configuring Tape and Medium Changer devices on IBM® System p models

Follow the same instructions as documented in the previous section. You must configure the Emulex Linux® device driver if you have Fibre Channel tape devices that are attached to your System p (also known as pSeries) system.

Configuring Tape and Medium Changer devices on IBM® System z models

The Fibre Channel topology that is supported for System z® is point-to-point and fabric. Refer to the Linux® on System z Fibre Channel documents for details on the supported configurations for Fibre Channel device attachment. The Linux Fibre Channel adapter device driver zfcp is available in the kernel that supports zSeries Fibre Channel Protocol. The zfcp device configuration methods in 2.6 (and higher) and 2.4 kernels are different. For 2.6 kernels and higher, refer to appropriate chapter in the Linux on System z document entitled *Linux on System z: Device Drivers, Features, and Commands*.

For 2.4 kernels, there are three ways to load the zfcp device driver to see the attached tape devices.

1. Create a /etc/zfcp.conf file and make a ram disk 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 that you restart the system, the zfcp is automatically loaded and the tape devices can be seen from the system. You must add the device map into this file. The following code is an example of zfcp.conf.

```
0xf1c0 0x1:0x5005076300402733 0x0:0x0000000000000000;\
0xf1c1 0x1:0x5005076300402733 0x0:0x0001000000000000
```

The zfcp 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

devno

The device number of the host bus adapter (16 bits, see /proc/subchannels). It is **0xf1c0** or **0xf1c1** in the previous 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 worldwide port name. It is **0x1** in the previous example.

wwpn

Worldwide port name that identifies the Fibre Channel port of the SCSI target device (64-bit). It is **0x5005076300402733** in the previous example.

unit scsi-lun

Linux internal SCSI Logical Unit Number (32-bit). It is **0x0** in the previous example.

fcplun

Logical Unit Number that is associated with the SCSI target device (64-bit). In the previous example, `0x0000000000000000` is the Logical Unit Number 0, and `0x0001000000000000` is the Logical Unit Number 1.

For tape attachment, each logical unit number must be associated with a unique devno. If you use the same devno numbers for several logical units, you can ensure that each <unit-scsi-lun> is unique. After `/etc/zfcp.conf` is created, run the following commands.

```
>mk_initrd>zipl
```

Then restart the system. After it is booted up, your tape device must be shown in `/proc/scsi/scsi` file.

2. Modify the `/etc/modules.conf` file to add the `zfcp` module parameters; then run the **depmod -A** and **modprobe zfcp** command.

Note: Do not use this choice together with the first one, otherwise it causes conflicts.

The `zfcp` map in `/etc/modules.conf` always takes higher priority than the map in `/etc/zfcp.conf`.

The following example demonstrates the `zfcp` configuration in `/etc/modules.conf`.

```
options zfcp map="\
0xf1c0 0x1:0x5005076300402733 0x0:0x0000000000000000;\
0xf1c1 0x1:0x5005076300402733 0x0:0x0001000000000000"
```

The map arguments are the same as the ones listed in for the `/etc/zfcp.conf` file.

After the `/etc/modules.conf` file is modified, save and close it. Then, run the following command.

```
>depmod -A
>modprobe zfcp
```

This action installs the `zfcp` device driver and all of its prerequisite kernel modules. Now you can check the file `/proc/scsi/scsi` to see whether 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.

Then, run the following commands to install `zfcp`.

```
>rmmod zfcp
>modprobe zfcp
```

3. Run the **modprobe zfcp** command first, then dynamically add a tape device into the system after you physically attach a Fibre Channel tape device to the switch. If you physically attach a tape device on the switch and `zfcp` is already loaded, you do not need to restart the Linux system to add this entry in the `/proc/scsi/scsi` file. The `zfcp` device driver provides an "add_map" proc system entry under the directory `/proc/scsi/zfcp` to dynamically add the device into the system. For example, to add two logical units from the example in Step 2 into the system, you can issue the following commands.

```
> echo "0xf1c0 0x1:0x5005076300402733 0x0:0x0000000000000000;\
0xf1c1 0x1:0x5005076300402733 0x0:0x0001000000000000"
```

```
> /proc/scsi/zfcp/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), and so on. 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 previous mapping. The value of `Lun` is the logical unit number of the target device, for example, the last number in the previous mapping. Currently, the `zfcp` device driver does not support dynamically removing the attached devices. If you must remove the tape devices from the system, do `rmmod zfcp`. Then, you can delete the entry in `/etc/modules.conf` and reload `zfcp`, or reload `zfcp` first and dynamically add the devices that you want. After you finished all the mapping, if you can see all of the attached tape devices in `/proc/scsi/scsi`, you successfully attached those devices to your system. Next, you can install the `lin_tape` device driver. Refer to the [Installation procedure](#) section for the instructions on how to install `lin_tape`.

Uninstallation procedure

Note: All tape devices that use the `lin_tape` driver must be closed and cannot be in use when `lin_tape` is uninstalled or the uninstall fails.
On SLES and RHEL:

Run the following command to uninstall `lin_taped` before you uninstall `lin_tape`:

```
> rpm -e lin_taped
```

Run the following command to uninstall `lin_tape`:

```
>rpm -e lin_tape
```

On Ubuntu:

```
apt purge lin-taped
apt purge lin-tape
```

Tape drive, media, and device driver parameters

This chapter describes the parameters that control the operating modes of the IBM® Linux® Tape and Medium Changer device driver.

- [Configuration parameters](#)
- [Nonchangeable parameters](#)
- [Changeable parameters](#)

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. These parameters are kept on reopen, but are always restored back to the default values when the lin_tape device driver is reinstalled.

Note: This procedure is completed with tape diagnostic and utility functions. See [IBM Tape Diagnostic Tool \(ITDT\)](#).

The nonchangeable configuration parameters are

- Autoloading
- Density code
- Emulate autoloader
- Hook word
- 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

- Barcode length
- Block size
- Buffered mode
- Capacity scaling
- Compression
- Disable auto drive dump
- Disable SIM logging
- Dynamic attributes
- Logging
- Logical write protect
- Maximum SCSI transfer length
- Read past filemark
- Rewind immediate
- Trace
- Trailer labels
- Busy Retry

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 as follows.

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 lin_tape.

Density code

This parameter is the density setting for the currently loaded tape. Some tape devices support multiple densities and report the current setting in this field. It cannot be changed by the application.

Emulate autoloader

This parameter currently is not supported and is ignored.

Hook word

This parameter is not supported in the lin_tape 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 lin_tape.

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 current loaded tape. Some tape devices support multiple media types and different values are reported in this field.

Read SILI bit

SILI bit currently is not supported due to limitations associated with the Linux® environment. SILI bit support can be enabled in future releases of the lin_tape.

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 lin_tape. 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. The lin_tape device driver ignores this field.

Write protect

This parameter is set to TRUE if the currently mounted tape is logically or physically write protected.

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 as follows.

Barcode length

This parameter can be set to change the barcode length for a cartridge. For LTO cartridges the default is 8. It can be changed to 6 for LTO 1 and LTO 2 generation cartridges only. For 3592 cartridges the default is set at 6. It can be changed to 8. In the /etc/modprobe.conf.local file the following line must be added to reflect the desired change.

```
options lin_tape ibm3592_barcode=8
options lin_tape lto_barcode=6
```

Block size

This parameter specifies the block size that is used for read and write operations. A value of zero means a variable block size. Any other value is a fixed block size. The installation default is zero (variable length block size). Refer to [Maximum SCSI transfer length](#) for guidance.

Buffered mode

This parameter specifies whether read and write operations must be buffered by the tape device. The default (recommended) value is TRUE.

Capacity scaling

This parameter sets the capacity or logical length of the current tape on Enterprise Tape System 3590 or 3592 tape subsystems. By reducing the capacity of the tape, the tape drive can access data faster at the expense of data capacity. Capacity scaling can be set at 100% for the entire tape (which is the default), or set at 75%, 50%, or 25% of the 3590 tape cartridge and more available capacity scaling for the 3592 standard 300 GB rewritable data cartridge. Capacity scaling remains with the tape across mounts until it is changed.

Note:

1. The tape position must be at the start of the tape to change this parameter from its current value.
2. Changing this parameter destroys any existing data on the tape.
3. For 3592 media types, capacity scaling is supported only for the standard 300 GB rewritable data cartridge. Attempting to set capacity scaling that is not supported by a device or the current loaded media always returns 100% and cannot be changed. For example, the 60 GB (Economy Data) cartridge for the IBM® 3592 cannot be capacity-scaled and is always 100%.

Compression

Hardware compression is implemented in the device hardware. This parameter turns the hardware 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).

Disable auto drive dump

This parameter is provided in the `lin_tape` version 1.2.2 or later. It is set to `FALSE` by default. If it is `FALSE` and the `lin_taped` daemon is running and if an error occurs in the drive that creates a drive dump, the `lin_tape` device driver automatically retrieves the drive dump and saves it under the `/var/log` directory by default. You can specify another directory in the `/etc/lin_taped.conf` file. The dump is labeled with a `.dmp` extension on the file. Refer to [Configuring and running the lin_taped daemon](#) for details.

Disable SIM logging

This parameter is provided in the `lin_tape` version 1.2.2 or later. It is set to `FALSE` by default. If it is `FALSE` and the `lin_taped` daemon is running and SIM/MIM data is generated by the drive, the `lin_tape` device driver automatically retrieves the data and saves it in a formatted text file under the `/var/log` directory by default. You can specify another directory in the `/etc/lin_taped.conf` file. Refer to [Configuring and running the lin_taped daemon](#) for details.

This capacity is not applicable to IBM Ultrium tape drives.

Dynamic attributes

This parameter determines whether dynamic runtime attributes are attempted on open for supported drives. Default is 1 (on) meaning that the driver automatically attempts to set dynamic runtime attributes on open. This parameter can be changed to 0 (off) in the configuration file before the `lin_tape` is loaded. It is recommended to keep on dynamic attributes unless it produces an unexpected problem in the environment.

Logging (volume logging)

This parameter turns the volume information logging On or Off. With the `lin_tape` version 1.2.2 and later, the `lin_tape` device driver provides this support. It is set to On by default. If logging is On and the `lin_taped` daemon is running, the `lin_tape` device driver retrieves the full log sense data from the drive whenever a tape is unloaded, or the drive reaches a log threshold. The log file is saved in binary format under the directory `/var/log` by default. You can specify another directory in `/etc/lin_taped.conf` file. Refer to [Configuring and running the lin_taped daemon](#) for details.

Note: This parameter is volume logging, which is different from error logging. `lin_tape` provides error logging whenever the `lin_taped` daemon is running. Refer to [Configuring and running the lin_taped daemon](#) for details on error logging.

Logical write protect

This parameter sets or resets the logical write protect of the current tape on Enterprise Tape System 3590 tape subsystems. The three types of logical write protect are associated protect, persistent protect, and write-once read-many (WORM) protect.

1. Associated protect remains only while the current tape is mounted or associated with the tape drive. It is reset when the tape is unloaded or the tape drive is reset.
2. Persistent protect remains or persists with the tape across mounts until it is reset.
3. WORM protect also remains with the tape across mounts, but unlike persistent protect it cannot be reset on the tape. After a tape is WORM protected, it can never be written on again.

Note: The tape position must be at the start of the tape to change this parameter from its current value.

Maximum SCSI transfer length

In the `lin_tape` drivers with level lower than 3.0.3, the maximum transfer length per device per SCSI command is 262144 bytes (256 KB) by default. Variable block read/write requests with transfer length greater than the maximum transfer length fails [*errno*: EINVAL]. When a fixed block size is 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 80000 bytes and maximum transfer length of 262144, a write request for 400000 bytes (5 blocks of 80000 each) is written to tape in two transfers. The first transfer is 240000 bytes (3 blocks) and the second transfer is 160000 (the remaining two blocks). You can increase the maximum transfer length to enhance the data throughput. This procedure can be done with ITDT with the Query/Set Parameters option, 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. Maximum transfer length of 256 KB is highly recommended.

In `lin_tape` driver with level 3.0.5 or higher and the open source driver `lin_tape`, the maximum transfer length is defined as the minimum length that the host bus adapter and the tape drive can support. This number is greater than 256 KB. It cannot be changed by the STIOCSETP *ioctl* call any more.

Read past filemark

If this parameter is set to true, when a fixed-length read operation encounters a filemark, it returns the number of bytes read before the filemark is encountered and positions the tape head after the filemark. If the `read_past_filemark` parameter is set to false, when the fixed-length read operation encounters a filemark, 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, then the *read* returns 0 bytes read and positions the tape head after the filemark.

This installation default is `FALSE`.

Rewind immediate

This parameter sets the immediate bit for rewind commands. If it is set to On, the rewind tape operation runs faster, but the next command takes a long time to finish unless the physical rewind operation is complete. 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 or Off. With the `lin_tape` version 1.2.2 and later, the `lin_tape` device driver provides this support. It is set to On by default. If trace is On and the `lin_taped` daemon is running, the `lin_tape` device driver retrieves the trace from the driver if trace level is set to 1 or 2 in the `/etc/lin_taped.conf` file. The trace file is saved under the directory `/var/log` by default. You can specify another directory in `/etc/lin_taped.conf` file. Refer to [Configuring and running the lin_taped daemon](#) for details.

Trailer labels

If this parameter is set to On, then 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 *errno* EIO is returned.

If this parameter is set to Off, then writing records past the early warning mark is not allowed. *Errno* variable is set to ENOSPC.

The installation default is On (with trailer labels).

Busy Retry

The parameter *busy_retry* determines how many times to retry a command when the device is busy. Default is 0 (off), and can be set up to 480 in the configuration file before the *lin_tape* is loaded.

lin_tape_ignoreOEM

The parameter *lin_tape_ignoreOEM* stops OEM devices from connecting to the driver. Default is 0 (off), and can be set up to 1 in the configuration file before the *lin_tape* is loaded.

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](#)
- [Special files for the Medium Changer device](#)
- [Persistent naming support](#)

Special files for the tape device

Each tape device has a set of special files providing access to the same physical drive but providing different attributes. [Table 1](#) shows the attributes of the special files.

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 1. Linux: Special files for IBM® 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 that is 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 that are designed for a tape device

Persistent naming support

lin_tape Persistent Naming is implemented through the Linux® *udev* utility. *Udev* is a service that monitors changes in system hardware configuration and completes actions that are based on what devices are attached to the Linux system. It can be configured to create symbolic links (persistent names) to a device based on attributes that a driver exports for that device. The persistent name can then be used as the device name to open and complete IO to a tape drive or medium changer. This action makes it possible to reference a static name, such as

```
/dev/lin_tape/by-id/lin_tape4801101
```

This name is always associated with the same physical device, rather than being required to reference the device name */dev/IBMTape0*, which can change names and become */dev/IBMTape1* after the driver is reinstalled.

lin_tape exports several attributes that can be used as the basis to create persistent names. These attributes can be reported to the user through *udevadm info* on recent Linux kernels, or *udevinfo* on older Linux kernels. The *udevinfo* and *udevadm* are *udev* management tools. These tools can query the *udev* database for device information

and properties that are stored in the udev base for help creating udev rules.

Note: Udev, udevinfo, and udevadm are not implemented or maintained by the lin_tape driver. Refer to the man pages for udevadm or udevinfo for details on usage. These man pages are system specific and supersede all information in this document. For questions on these utilities, you must contact your Linux support representative. An example is provided on udev for implementing a persistent name. The example must be customized to fit a user's needs and environment.

Note: Variations exist between kernels.

If a tape device is attached to the Linux system with worldwide port name **0x123456789ABCDEF0** with a current device name of **/dev/IBMtape0**, a user can run udevadm information to obtain information on exported attributes for this device. This procedure can be done as follows,

```
>udevadm info --attribute-walk --name /dev/IBMtape0
```

The output of this command includes something similar to

```
ATTRS{serial_num}=="123456789"
ATTRS{ww_node_name}=="0x123456789ABCDEF1"
ATTRS{ww_port_name}=="0x123456789ABCDEF0"
```

If you are using udevinfo, enter the previous command as

```
>udevinfo -a -p `udevinfo -q path -n /dev/IBMtape0`
or
>udevinfo -a -p /sys/class/lin_tape/IBMtape0
(Both commands return the same information).
```

Also, on some kernels an attribute **ATTRS{xxx}** is replaced by **SYSFS{xxx}**. Furthermore, some kernels use a '=' (single equal sign) to indicate an attribute match and also an assignment. Whereas, other kernels use a '==' (double equal sign) for a match and '=' for assignment. Place the attribute from the attribute list into your rules file exactly as it appears in the attribute list, as described here.

The **ww_port_name** is in a rules file that assigns a symbolic link to a device that has the listed worldwide port name. The file typically is placed in **/etc/udev/rules.d**, but this location might be changed by the **udev_rules** directive in the **/etc/udev/rules.conf** file. In this example, a file is created that is called **/etc/udev/rules.d/98-lin_tape.rules** and write a single line to the file.

```
KERNEL=="IBMtape*", ATTRS{ww_port_name}=="0x123456789ABCDEF0",
SYMLINK="lin_tape/by-id/lin_tape4801101"
```

Assuming that the udev service is running and configured correctly, the user can install or reinstall **lin_tape** with **modprobe**, and the symbolic link is created in the **/dev/lin_tape/by-id** folder. One line must be added to the **98-lin_tape.rules** file for each wanted symbolic link.

With **lin_tape** version 2.2.0, a parameter, **persistent_n_device**, is added to support persistent naming of no rewind devices. The default value is 0 (off). To enable this support, complete the following steps.

1. Modify the **98-lin_tape.rules** file to differentiate standard devices and no rewind devices. For example,

```
KERNEL=="IBMtape*[0-9]", ATTR{serial_num}=="1013000306", SYMLINK="lin_tape/by-id/IBMtape0"
KERNEL=="IBMtape*n", ATTR{serial_num}=="1013000306", SYMLINK="lin_tape/by-id/IBMtape0n"
```

2. Stop the **lin_taped** daemon.

```
lin_taped stop
```

3. Unload the **lin_tape** driver from the memory.

```
modprobe -r lin_tape
```

```
Only if sfmp was enabled unload stmp/sgmp:
modprobe -r stmp
modprobe -r sgmp
```

```
modprobe -r pfo
```

4. Add the following line in your **/etc/modprobe.conf** or **/etc/modprobe.conf.local** file (or, if you are running RHEL 6 or higher, in your **/etc/modprobe.d/lin_tape.conf** file).

```
options lin_tape persistent_n_device=1
```

5. Reload the **lin_tape** driver into memory.

```
modprobe pfo
modprobe lin_tape
```

```
Only if sfmp was enabled load stmp/sgmp:
modprobe stmp
modprobe sgmp
```

Note: Wait at least 10 seconds between step 3 (**modprobe -r lin_tape**) and step 5 (**modprobe lin_tape**) in order for udev to correctly configure the devices.

6. Check that the devices are all correctly listed with the following command.

```
ls -l /dev/lin_tape/by-id/
```

7. Restart the **lin_taped** daemon.

```
lin_taped start
```

Control Path failover support for tape libraries

Note: The library control path failover feature code must be installed before control path failover support is enabled in the Linux® lin_tape device driver. Refer to [Automatic failover](#) to determine which feature code is required for your machine type.

The Linux lin_tape device driver control path failover support configures multiple physical control paths to the same logical library within the device driver. It also provides automatic failover to an alternative control path when a permanent error occurs on one path. This support is transparent to the running application.

- [Configuring and unconfiguring path failover support](#)
- [Enabling path failover for the sg interface](#)
- [Primary and alternative paths](#)
- [Querying primary and alternative path configuration](#)
- [Disabling and enabling primary and alternative paths](#)

Configuring and unconfiguring path failover support

Control path failover support is not enabled automatically when the device driver is installed. The Linux® lin_tape device driver provides a driver parameter `alternate_pathing` for you to enable the library control path failover. To enable the failover support in the lin_tape device driver software, you must do the following steps after the lin_tape rpm package is installed:

1. `lin_taped stop` (stop the lin_taped daemon)
2. Unload the lin_tape driver from the memory.

```
modprobe -r lin_tape
```

```
Only if sfmp was enabled unload stmp/sgmp:  
modprobe -r stmp  
modprobe -r sgmp
```

```
modprobe -r pfo
```

3. Add the following line in your `/etc/modprobe.conf` or `/etc/modprobe.conf.local` file (or, if you are running RHEL 6 or higher, in your `/etc/modprobe.d/lin_tape.conf` file)

```
options lin_tape alternate_pathing=1
```

4. Reload the lin_tape driver into memory.

```
modprobe pfo  
modprobe lin_tape
```

```
Only if sfmp was enabled load stmp/sgmp:  
modprobe stmp  
modprobe sgmp
```

5. `lin_taped start` (restart lin_taped daemon)

You can ignore the `Unresolved symbols in /lib/modules/<your kernel name>/drivers/scsi/lin_tape.ko` message after the `depmod` command. You can check whether the lin_tape driver 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 successfully enabled the control path failover feature for your library. If "NA" is listed under "FO Path", then the control path failover is not enabled. After control path failover support is enabled, it remains set until the lin_tape driver is reloaded with the `alternate_pathing` driver parameter set to OFF. The path failover setting is retained even if the system is rebooted. If you want to turn off the control path failover feature in the lin_tape device driver, you can complete the following steps.

1. `lin_taped stop`
2. Unload the lin_tape driver from the memory.

```
modprobe -r lin_tape
```

```
Only if sfmp was enabled, unload stmp/sgmp:  
modprobe -r stmp  
modprobe -r sgmp
```

```
modprobe -r pfo
```

3. Delete the following line in your `/etc/modprobe.conf` file.

```
options lin_tape alternate_pathing=1
```

4. Reload the lin_tape driver into memory.

```
modprobe pfo  
modprobe lin_tape
```

```
Only if sfmp was enabled load stmp/sgmp:  
modprobe stmp  
modprobe sgmp
```

5. `lin_taped start` (restart lin_taped daemon)

Enabling path failover for the sg interface

Through a collaboration effort that is named join driver, control path failover is supported through the `sg` interface, by using the latest `lin_tape` driver version on RHEL and SLES over Intel only.

Note: It is important to review hardware and software requirements before path failover is enabled. See [Path failover and load balancing](#).

To enable it, the `lin_tape_as_sfmp` parameter must be set at `/etc/modprobe.d/lin_tape.conf` as follows:

```
options lin_tape alternate_pathing=1
options lin_tape lin_tape_as_sfmp=1
```

If `lin_tape` is already installed, follow the [Uninstallation procedure](#).

Follow the [Installation procedure](#) by using

```
-with sfmp at rpmbuild.
```

Primary and alternative paths

When `lin_tape` is loaded into kernel memory, the first logical medium changer device that `lin_tape` sees in the system is the primary path for that medium changer. The other logical medium changers that `lin_tape` attached for the same medium changer are configured as alternative paths. The device driver supports up to 16 physical paths for a single device. The primary and alternative path information can be obtained by the following command.

- `cat /proc/scsi/IBMchanger`

An example of a `/proc/scsi/IBMchanger` file:

- `lin_tape` version: 3.0.3
- `lin_tape` major number: 253

Table 1. Attached changer devices

Number	Model	SN	HBA	FO Path
0	03584L22	IBM1234567	qla2xxx	Primary
1	03584L22	IBM1234567	qla2xxx	Alternate
2	03584L22	IBM1234567	qla2xxx	Alternate

The labeling of a logical device as either a primary or alternative path is for information only, to

- Identify the actual number of physical devices that are configured on the system and a specific logical device that is associated with them. Only one logical device is labeled as the primary path for each physical device. However, multiple logical devices can be labeled as an alternative path for the same devices.
- Provide information about which logical devices configured on the system have path failover support enabled.

The numbers listed in [Table 1](#) are the ones used for `IBMchanger` special files at `/dev` directory (see [Special Files for the Medium Changer device](#)). An attempt to open a device file name not listed at `/dev` directory will fail. Per file systems handling and due to path removal, usually a device file name is deleted after a device closes or before a device opens. Using [Persistent naming support](#) will maintain device file names listed under `ls -l /dev/lin_tape/by-id/`.

When `lin_tape_as_sfmp` is set, `sg` paths can be queried through `pfo` paths as follows:

```
cat /sys/class/pfo/*/paths
```

Example output:

```
pfo10 sg=/dev/sg10 st=none sf=yes fo=no wwnn=0000013400140405 type=changer
2:0:1:1 up last - -
3:0:1:1 - - - -
```

There is only one `sg` device file name per device that uses all the paths for this device.

Querying primary and alternative path configuration

You can show the primary and alternative path configuration for all devices by reading the `/proc/scsi/IBMchanger` file, as explained in [Primary and alternative paths](#). Note: Show the primary and alternative path configuration for any device with tape diagnostic and utility functions. Refer to [IBM Tape Diagnostic Tool \(TDT\)](#).

Disabling and enabling primary and alternative paths

When you load the `lin_tape` device driver with the `alternate_pathing` parameter to be ON, by default, all the available paths for a physical device are enabled.

If it is necessary to disable a path and not run path failover (for example, because of maintenance), run commands to disable and then later enable the primary and alternative paths.

The commands to enable and disable primary and alternative paths are tape diagnostic and utility functions.

Note: See [IBM Tape Diagnostic Tool \(TDT\)](#).

Data Path failover and load balancing support for tape drives

Data path failover support is not enabled automatically when the device driver is installed. The Linux® `lin_tape` device driver provides a driver parameter `alternate_pathing` for you to enable the data path failover.

To enable the failover support in the lin_tape device driver software, you must complete the following steps after the lin_tape rpm package is installed.

```
>lin_taped stop (stop the lin_taped daemon)
>rmmod lin_tape (unload the lin_tape driver from the memory)
```

If you have IBM® 3592 tape drives, add the following line in your /etc/modprobe.conf or /etc/modprobe.conf.local file (or, if you are running RHEL 6 or higher, in your /etc/modprobe.d/lin_tape.conf file).

```
options lin_tape alternate_pathing=1
```

If you have IBM LTO tape drives, the library must have a path failover feature code. The data path failover license keys are needed to enable the failover if you are running LTO2 drives or if you are running LTO3 drives with old levels of drive code. DPF keys do not need to be added if you are running the latest drive code on LTO3 or higher drives.

Add the following line in your /etc/modprobe.conf or /etc/modprobe.conf.local file (or, if you are running RHEL 6 or higher, in your /etc/modprobe.d/lin_tape.conf file).

```
options lin_tape 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 lin_tape (re-load the lin_tape driver into memory)
>lin_taped (re-start lin_taped daemon)
```

You can ignore the Unresolved symbols in /lib/modules/<your kernel name>/drivers/scsi/lin_tape.ko message after the **depmod** command. You can check whether the lin_tape driver 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 successfully enabled data path failover feature for your tape drive. If NA is listed under FO Path, the data path failover is not enabled. After the path failover support is enabled, it remains set until the lin_tape driver is reloaded with the alternate_pathing driver parameter set to OFF. The path failover setting is retained even if the system is rebooted. If you want to turn off the data path failover feature in the lin_tape device driver, you can run the following steps.

```
>lin_taped stop
modprobe -r lin_tape
```

Only if sfmp was enabled unload stmp/sgmp:

```
modprobe -r stmp
modprobe -r sgmp
```

```
modprobe -r pfo
```

Delete the following line in your /etc/modules.conf file: **options lin_tape alternate_pathing=1.**

```
modprobe pfo
modprobe lin_tape
```

Only if sfmp was enabled load stmp/sgmp:

```
modprobe stmp
modprobe sgmp
```

```
>lin_taped start
```

- [Enabling path failover for st and sg interface](#)
- [Primary and alternative paths](#)
- [Querying primary and alternative path configuration](#)
- [Disabling and enabling primary and alternative paths](#)
- [Tape Reserve Type](#)

Enabling path failover for st and sg interface

Through a collaboration effort that is named join driver, control path failover is planned to be supported through st and sg interfaces by using the latest lin_tape driver version on RHEL and SLES over Intel only.

Important: Review hardware and software requirements before path failover is enabled. See [Path failover and load balancing](#).

To enable it, the lin_tape_as_sfmp parameter must be set at /etc/modprobe.d/lin_tape.conf:

```
options lin_tape alternate_pathing=1
options lin_tape lin_tape_as_sfmp=1
```

If lin_tape is already installed, follow the [Uninstallation procedure](#).

Follow [Installation procedure](#) by using

```
-with sfmp at rpmbuild.
```

Primary and alternative paths

When the `lin_tape` device driver is loaded into kernel memory with path failover support enabled, the first logic device that `lin_tape` sees always becomes the primary path. The other logical devices that `lin_tape` sees are configured as the alternative paths. The device driver supports up to 16 physical paths for a single device.

The primary and alternative path information can be obtained by the following command.

```
>cat /proc/scsi/IBMtape
```

The following is an example of a `/proc/scsi/IBMtape`:

```
lin_tape version: 3.0.3
lin_tape major number: 253
```

Table 1. Attached tape devices

Number	Model	SN	HBA	FO Path
0	03592	IBM1234567	qla2xxx	Primary
1	03592	IBM1234567	qla2xxx	Alternate

The labeling of a logical device as either a primary or alternative path is for information only and is used for following purpose:

- Identify the actual number of physical devices that are configured on the system and a specific logical device that is associated with them. Only one logical device is labeled the primary path for each physical device. However, many (multiple) logical devices can be labeled as an alternative path for the same device.
- Provide information about which logical devices configured on the system have path failover support enabled.

The numbers listed in [Table 1](#) are the ones used for IBMchanger special files at `/dev` directory (see [Special files for the tape device](#)). An attempt to open a device file name not listed at `/dev` directory will fail. Per file systems handling and due to path removal, usually a device file name is deleted after a device closes or before a device opens. Using [Persistent naming support](#) will maintain device file names listed under `ls -l /dev/lin_tape/by-id/`.

When `lin_tape_as_sfmp` is set, `st` and `sg` paths can be queried through `pfo` paths as follows:

```
cat /sys/class/pfo/*/paths
```

Example output:

```
pfo9 sg=/dev/sg9 st=/dev/st0 sf=yes fo=no wwnn=500507630f04be07 type=tape
2:0:1:0 up last - wwpn=500507630f44be07
3:0:1:0 - - -
```

Only one `st` and one `sg` device file name per device can use all the paths for that device.

Querying primary and alternative path configuration

You can show the primary and alternative path configuration for all devices by reading the `/proc/scsi/IBMtape` file, as explained in [Primary and alternative paths](#). Note: Show the primary and alternative path configuration for any device with tape diagnostic and utility functions. Refer to [IBM Tape Diagnostic Tool \(TDT\)](#).

Disabling and enabling primary and alternative paths

If it is necessary to disable a path and not run path failover (for example, because of maintenance), run commands to disable and then enable the primary and alternative paths.

The commands to enable and disable primary and alternative paths are tape diagnostic and utility functions. Note: See [IBM Tape Diagnostic Tool \(TDT\)](#).

Tape Reserve Type

This parameter causes `lin_tape` to issue SCSI-3 persistent reserves to a tape drive whenever a reservation is attempted. Persistent reserves are automatically issued if data path failover is used, and therefore setting the parameter is unnecessary. This parameter can be set only when `lin_tape` is installed. To set it, add the following line to `/etc/modprobe.conf` or `/etc/modprobe.conf.local` (or, if you are running RHEL 6 or higher, to `/etc/modprobe.d/lin_tape.conf`):

```
options lin_tape tape_reserve_type=persistent
```

If alternate pathing is not enabled persistent reserve is not issued automatically, so this parameter is needed only if you want to use persistent reserve with `alternate_pathing` disabled.

Open source device driver - lin_tape

The `lin_tape` device driver is the new device driver for the Linux® 2.6 kernels to replace the closed-source driver `IBMtape`. In most respects, it behaves the same as the closed-source `IBMtape` device driver. This section covers significant differences between the `IBMtape` driver and the `lin_tape` driver.

- [Comparing IBMtape and lin_tape](#)
- [Installation](#)
- [Driver parameters and special device files](#)
- [Taking devices offline and completing maintenance](#)
- [Path failover support](#)
- [lin_taped daemon](#)

Comparing IBMtape and lin_tape

[Table 1](#) compares the names for various components of the IBMtape and lin_tape device drivers.

Table 1. Comparing IBMtape and lin_tape

Component	IBMtape	Lin_tape
Driver name	IBMtape	lin_tape
Module name	IBMtape.ko	lin_tape.ko
Special files	/dev/IBMtape0 /dev/IBMchanger0, and so on.	No change
Proc entry	/proc/scsi/IBMtape /proc/scsi/IBMchanger	No change
Daemon name	IBMtaped	lin_taped
Daemon configuration file	/etc/IBMtaped.conf	/etc/lin_taped.conf
Daemon trace files	/var/log/IBMtape.trace /var/log/IBMtape.errorlog	/var/log/lin_tape.trace /var/log/lin_tape.errorlog

Lin_tape join driver installs pfo.ko, an extra lin_tape module that is idle for non-sfmp configurations, and blacklist-pfo.conf at the location /etc/modprobe.d for pfo module load control.

Installation

Installation of the lin_tape driver is the same as for the IBMtape driver, except that IBMtape must be replaced with lin_tape in the installation instructions. Refer to [Installation and Configuration instructions](#) for details.

The lin_tape driver cannot be installed if the IBMtape driver is already installed. If the IBMtape driver is installed, first uninstall the IBMtape driver, and then install the lin_tape driver. With RHEL4 and SLES10, driver removal also requires a reboot of the server, since the IBMtape driver module is "permanent" in these distributions.

Driver parameters and special device files

The driver parameters are not changed for the lin_tape driver. However, it is important that the module parameters, such as alternate_pathing and dpf_keys, must now be applied to the lin_tape module, instead of the IBMtape module. For example, in the /etc/modprobe.conf or /etc/modprobe.conf.local file (or, if you are running RHEL 6 or higher, the /etc/modprobe.d/lin_tape.conf file), add the following line for LTO library's path failover:

```
options lin_tape alternate_pathing=1 dpf_keys="abcdefghijklmnp"
```

abckdefghijklmnp is an example of a data path failover feature key.

The special device files for the lin_tape driver are the same as for the IBMtape driver. Refer to [Special files for the tape device](#) and [Special files for the Medium Changer device](#) for details on special device files.

Taking devices offline and completing maintenance

Input and output must be quiesced and all driver handles must be closed before a lin_tape device is taken offline. It is recommended to remove the lin_tape driver module and then the HBA driver module before maintenance is done or the physical topology of the tape drive or library environment is changed. Lin_tape can be removed by the following command at the shell prompt.

```
lin_taped stop
modprobe -r lin_tape
```

Only if sfmp was enabled unload stmp/sgmp:

```
modprobe -r stmp
modprobe -r sgmp
```

```
modprobe -r pfo
```

Likewise, the HBA module can be removed by the following command, where "HBA_driver" is your specific HBA driver.

```
rmmmod <HBA_driver>
```

After the maintenance is complete and the environment is ready to issue input and output, the drivers must be reinstalled in reverse order. This procedure is typically done by

```
modprobe <HBA_driver>
modprobe pfo
modprobe lin_tape
```

Only if sfmp was enabled load stmp/sgmp:

```
modprobe stmp
modprobe sgmp
```

```
>lin_taped start
```

Path failover support

Path failover support in `lin_tape` is the same. However, with the `lin_tape` driver, failover support is provided through the `lin_taped` daemon. If the `lin_taped` daemon is not running, neither control path failover nor data path failover is attempted. The `lin_taped` daemon is started automatically when the `lin_tape` driver is loaded.

To check whether the `lin_taped` daemon is running, run the following command.

```
lin_taped status
```

This command indicates whether the `lin_taped` daemon is running. If the `/proc/scsi/IBMtape` and `/proc/scsi/IBMchanger` files indicate "NA" for "FO Path", this answer indicates that failover support for that device is not enabled. If all other settings are correct, but "FO Path" is incorrectly indicating "NA", confirm that the `lin_taped` daemon is running.

For details about path failover support, refer to [Control Path failover support for tape libraries](#) and [Data Path failover and load balancing support for tape drives](#).

lin_taped daemon

The `lin_taped` daemon uses the same command-line arguments as the `IBMtaped` daemon. The `lin_taped` configuration file is the same as the `IBMtaped` configuration file, but is renamed to `lin_taped.conf`. Refer to [Configuring and running the lin_taped daemon](#) for information.

System-managed encryption

- [Configuring device drivers](#)
- [Querying tape drive configuration](#)

Configuring device drivers

Note: System-managed encryption (SME) on Linux® requires that the `lin_taped` daemon is running.

The device driver SME settings can be set for all drives at once with the `default_sys_encryption_proxy` and `default_sys_encryption_write` module options.

If no options are specified in the registry, the driver uses the default values for the parameters.

- The default value for `default_sys_encryption_proxy` is 1.
This value causes the device driver to handle encryption key requests, if the drive is set up for system-managed encryption. This value does not need to be changed. A value of 0 causes the device driver to ignore encryption key requests for system-managed encryption drives, and is not desirable.
- The default value for `default_sys_encryption_write` is 2.
This value causes the device driver to leave the encryption write-from-BOP settings alone. It does not turn on or turn off encryption writing, but instead uses the settings that are already in the drive. If encryption is not set up previously, then the drive writes unencrypted data. A value of 0 causes the device driver to write unencrypted data. A value of 1 causes the device driver to write encrypted data.

The module options can be specified in the `/etc/modprobe.conf`, `/etc/modprobe.conf.local`, or `/etc/modprobe.d/lin_tape.conf` files, the same as other `lin_tape` module parameters.

For example, to turn on SME to write/read encrypted data,

1. Add the following line:

```
options lin_tape default_sys_encryption_write=1
```

2. Then, run the following commands.

```
>lin_taped stop  
>modprobe -r lin_tape  
  
Only if sfmp was enabled unload stmp/sgmp:  
>modprobe -r stmp  
>modprobe -r sgmp  
  
>modprobe -r pfo  
Wait at least 5 seconds  
>modprobe pfo  
>modprobe lin_tape  
  
Only if sfmp was enabled load stmp/sgmp:  
>modprobe stmp  
>modprobe sgmp  
  
>lin_taped start
```

The default settings are used to initialize the settings for all connected drives.

To modify the settings for individual drives, the settings are `sys_encryption_write` and `sys_encryption_proxy`. They have the same definitions and values as the similarly named "default" parameters, except that the settings apply only to individual drives.

These settings are available as part of the sysfs infrastructure. For each drive, there are two files, named `sys_encryption_write` and `sys_encryption_proxy`, in the `/sys/class/lin_tape/{DEVICE}/` directory, where: `{DEVICE}` is the device name, such as `IBMtape0`. The contents of these files indicate the current setting for the parameter

for that particular drive. The setting can be changed by writing a different value for the parameter to the file.

For example, to change the `sys_encryption_write` setting for IBMtape0 to ON (which has a value of 1), enter the following at a command line.

```
echo 1">/sys/class/lin_tape/IBMtape0/sys_encryption_write
```

Note: The driver encryption parameters for individual drives are not persistent between loads of the `lin_tape` driver. If you remove the `lin_tape` driver, and then reload it, the individual settings for all drives are the same as the "default" settings in `/etc/modprobe.conf`, `/etc/modprobe.conf.local`, or `/etc/modprobe.d/lin_tape.conf`. The `ibmekm.conf` file, which contains the configuration that the EKM servers use, is installed in the `/etc/` directory when `lin_tape` is installed. Instructions for modifying this file are found within the file itself.

Querying tape drive configuration

The following is an example of the output when the drive is configured for system-managed encryption, with encryption turned on.

```
issuing query encryption status...
encryption capable.....Yes
encryption method.....METHOD_SYSTEM
encryption state.....ON
```

Problem determination

A set of tools is provided with the device driver to determine whether the device driver and the tape device are functioning correctly.

- [Tracing driver modules](#)
- [Configuring and running the lin_taped daemon](#)
- [Reservation conflict logging](#)
- [Devices not reported at /proc/scsi/IBMchanger or /proc/scsi/IBMtape](#)

Tracing driver modules

By default, the driver prints minimal kernel trace messages to the system log at `/var/log/messages`. The minimal information includes notification that a device is recognized or taken offline and also the most serious of error conditions. If a more verbose trace is wanted, the variable `/sys/bus/scsi/drivers/lin_tape/lin_tape_debug` must contain the value 1. This procedure can be accomplished in one of two ways -

- Add the following line to `/etc/modprobe.conf`, `/etc/modprobe.conf.local`, or `/etc/modprobe.d/lin_tape.conf`.

```
options lin_tape lin_tape_debug=1
```

Then reinstall `lin_tape`. This action causes the `lin_tape_debug` variable to be set every time `lin_tape` is loaded.

- Issue the following command from the shell.

```
echo 1 > /sys/bus/scsi/drivers/lin_tape/lin_tape_debug
```

This action causes the `lin_tape_debug` variable to be set only until `lin_tape` is uninstalled or until the variable is set back to 0.

Configuring and running the lin_taped daemon

Starting with `lin_tape` version 1.2.5, the `lin_tape` device driver provides an error diagnostic daemon (`lin_taped`) which provides the following capabilities:

1. Error logging and tracing
2. When drive dumps, log sense data, or SIM/MIM error information is created by the tape drive, the daemon automatically retrieves that data and saves it to the hard disk drive on your Linux® system.
3. Failover and load balancing
4. Encryption

Because `lin_taped` requires a minimal amount of system resource and because it provides these necessary diagnostic capabilities, IBM® recommends that you leave the daemon always enabled.

Installing lin_taped

`lin_taped` is automatically installed at `/usr/bin/lin_taped` when you install the `lin_tape` device driver with the rpm or tar package. Refer to [Installation and Configuration instructions](#) for instructions on installing the `lin_tape` device driver.

Configuring lin_taped

You can customize the operation of `lin_taped` by modifying its configuration file, which is at `/etc/lin_taped.conf`. The daemon reads only the configuration file when it starts; so if you modify the configuration file, stop the daemon, and restart it so that your modifications are recognized by the daemon.

Tracing:

Three levels of tracing are supported for the `lin_taped` daemon. `lin_taped` tracing is a complement to, but is different from, tracing of the kernel module that is described in [Tracing driver modules](#). The `lin_taped` tracing levels are defined as follows:

0	With tracing set to 0, <code>lin_taped</code> records minimal tracing.
1	With tracing set to 1, <code>lin_taped</code> records information that is associated with each <code>ioctl</code> called. If a device error occurs and SCSI sense data is obtained from the device, a subset of that sense data is also recorded. The default setting for tracing.
2	With tracing set to 2, <code>lin_taped</code> records tracing messages for each SCSI command. If a device error occurs and SCSI sense data is obtained from the device, all sense data is also recorded. This tracing level is used only when a specific problem is being diagnosed due to the potential for huge amounts of data that is generated.

Set the `lin_tapeTrace` variable in the `/etc/lin_taped.conf` file to 0, 1, or 2, depending on what level of tracing you want. If the `lin_tapeTrace` variable is set to an invalid number, the `lin_taped` daemon does not start.

Tracing information is written to a file named `/var/log/lin_tape.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 `lin_tape.a` in the same directory. In the archive, the file name is renamed to `lin_tape.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 `lin_taped.conf` file. Refer to the instructions in the `lin_taped.conf` file for details.

Error logging:

`lin_taped` records certain error messages from the `lin_tape` device driver in a file named `/var/log/lin_tape.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 (with the Linux `ar` command) into file `lin_tape.a` in the same directory. In the archive, the file name is renamed to `lin_tape.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 `lin_taped.conf` file. Refer to the instructions in the `lin_taped.conf` file for details.

Whenever the `lin_taped` 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 4 bytes represent `driver_byte`, `host_byte`, `msg_byte`, and `status_byte`. "Sense Data" is the full SCSI sense data that is returned from the target. "Description" is a person-readable text string that is obtained by parsing the sense key field of the sense data.

The following circumstances are not logged in the `lin_tape.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 that 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 `lin_tape` 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 `lin_tape.timestamp.log`, where: *lin_tape* is the device special file (for example, `lin_tape1`, `lin_tape2`) and *timestamp* reflects the time that 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/lin_taped.conf` file.

There are two configuration parameters in the `/etc/lin_taped.conf` file that you can tailor to affect the number of log sense files that are kept on your system.

- `lin_tapeMaxLogSenseFiles`, which has a value of 0 or a positive decimal number.
- `lin_tapeAutoLogSenseFileOverWrite`, which has a value of 0 or 1.

By default, `lin_tapeMaxLogSenseFiles` is 0 and `lin_tapeAutoLogSenseFileOverWrite` is 1, which means that every time log sense data is created, it is written to a new file.

If `lin_tapeMaxLogSenseFiles` is 0, `lin_tapeAutoLogSenseFileOverWrite` is ignored, and each time log sense data is obtained, it is written to a new file.

If `lin_tapeMaxLogSenseFiles` is a positive number and `lin_tapeAutoLogSenseFileOverWrite` is 0, each time log sense data is created, `lin_taped` writes that data to a file until `lin_tapeMaxLogSenseFiles` is created. Then, `lin_taped` stops creating new files, even if new log sense data is produced.

If `lin_tapeMaxLogSenseFiles` is a positive number and `lin_tapeAutoLogSenseFileOverWrite` is 1, each time log sense data is created, `lin_taped` writes that data to a file until `lin_tapeMaxLogSenseFiles` is created. Then, when new log sense data is detected, `lin_taped` deletes the oldest log sense file and creates a new file with the newest log sense data. Thus, only the newest data is kept.

Automatically retrieving a drive dump:

If a condition occurs in the drive such that a drive dump is created, `lin_taped` retrieves the drive dump and saves it in a file named `lin_tapex.timestamp.dmp`, where *lin_tape* is the device special file (for example, `lin_tape1`, `lin_tape2`) and *timestamp* reflects the time that the file was created. Each time a drive dump is obtained, it is written to a new file. The IBM service organization 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/lin_taped.conf` file.

There are two configuration parameters in the `/etc/lin_taped.conf` file that you can tailor to affect the number of drive dumps that are kept on your system.

- `lin_tapeMaxDumpFiles`, which can have a value of 0 or a positive decimal number.

- `lin_tapeAutoDriveDumpFileOverWrite`, which can have a value of 0 or 1.

By default, `lin_tapeMaxDumpFiles` is 0 and `lin_tapeAutoDriveDumpFileOverWrite` is 1, which means that every time a drive dump is obtained, it is written to a new file.

If `lin_tapeMaxDumpFiles` is 0, `lin_tapeAutoDriveDumpFileOverWrite` is ignored, and each time a drive dump is obtained, it is written to a new file.

If `lin_tapeMaxDumpFiles` is a positive number and `lin_tapeAutoDriveDumpFileOverWrite` is 0, each time a dump is obtained, `lin_taped` writes that data to a file until `lin_tapeMaxDumpFiles` is created. Then, `lin_taped` stops creating new files, even if new drive dumps are produced.

If `lin_tapeMaxDumpFiles` is a positive number and `lin_tapeAutoDriveDumpFileOverWrite` is 1, each time a dump is obtained, `lin_taped` writes that data to a file until `lin_tapeMaxDumpFiles` is created. Then, when a new drive dump is detected, `lin_taped` deletes the oldest drive dump file and creates a new file with the newest drive dump data. Thus, only the newest data is kept.

Automatically retrieving SIM/MIM data:

If a condition occurs in the drive such that a drive SIM/MIM data is created, `lin_taped` retrieves the data and save it in a file named `lin_tapex.timestamp.simmim`, where `lin_tapex` is the device special file (for example, `lin_tape1`, `lin_tape2`) and timestamp reflects the time that the file was created. Each time SIM/MIM data is obtained, it is written to a new file. The IBM service organization might request that you forward SIM/MIM data to them for analysis.

The SIM/MIM data is stored in the `/var/log` directory by default. You can specify another directory in the `/etc/lin_taped.conf` file.

There are two configuration parameters in the `/etc/lin_taped.conf` file that you can tailor to affect the number of SIM/MIM files that are kept on your system.

- `lin_tapeMaxSimMimDataFiles`, which can have a value of 0 or a positive decimal number.
- `lin_tapeAutoSimMimDataOverWrite`, which can have a value of 0 or 1.

By default, `lin_tapeMaxSimMimDataFiles` is 0 and `lin_tapeAutoSimMimDataOverWrite` is 1, which means that every time SIM/MIM data is obtained, it is written to a new file.

If `lin_tapeMaxSimMimDataFiles` is 0, `lin_tapeAutoSimMimDataOverWrite` is ignored, and each time SIM/MIM data is obtained, it is written to a new file.

If `lin_tapeMaxSimMimDataFiles` is a positive number and `lin_tapeAutoSimMimDataOverWrite` is 0, each time SIM/MIM data is obtained, `lin_taped` writes that data to a file until `lin_tapeMaxSimMimDataFiles` is created. Then, `lin_taped` stops creating new files, even if new SIM/MIM data is created.

If `lin_tapeMaxSimMimDataFiles` is a positive number and `lin_tapeAutoSimMimDataOverWrite` is 1, each time SIM/MIM data is obtained, `lin_taped` writes that data to a file until `lin_tapeMaxSimMimDataFiles` is created. Then, when new SIM/MIM data is detected, `lin_taped` deletes the oldest SIM/MIM file and creates a new file with the newest SIM/MIM data. Thus, only the newest data is kept.

Selective tracing:

`lin_tape` 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 an application program, which 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 `lin_tape` 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 controls the logging of SIM/MIM data for a device. By default it is set to Off, which causes SIM/MIM data to be logged. Set this flag to On to suppress the logging of SIM/MIM records.

`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.

Running `lin_taped`:

If you are running the `lin_tape` device driver, version 1.4.1 or higher, after installing `lin_tape` `lin_taped` starts running even if your system does not have a tape device attached. If you add a new tape device into your Linux system, `lin_taped` automatically creates a special file under the `/dev` directory. If you are running the `lin_tape` device driver, version 1.3.x or less, `lin_taped` does not automatically start if there is no tape device attached. After you attach a new tape device, you must start the `lin_taped` daemon.

You can start `lin_taped` from the command line. `lin_taped` takes zero or more of the parameters as listed in the following command.

`lin_taped [start stop restart status]`

`lin_taped` or `lin_taped start`

Starts the daemon. If there is already a `lin_taped` running, the new one is aborted. (Use `lin_taped restart` if `lin_taped` is already running.)

`lin_taped stop`

Terminates the daemon and frees all the resources that are associated with the daemon. When the daemon is stopped, no information is saved.

`lin_taped restart`

Terminates the currently running daemon and starts a new one. The new daemon reads the `/etc/lin_taped.conf` file. This command is used after the `/etc/lin_taped.conf` file is modified while `lin_taped` is running.

`lin_taped status`

Prints a message on stdout to indicate whether the daemon is running or not.

Note: If you run `rmmod lin_tape` command to remove the `lin_tape` device driver from the running kernel, you must stop the `lin_taped` daemon first; otherwise you get a Device or Resource Busy error.

Reservation conflict logging

When the device driver receives a reservation conflict on a tape drive command, it logs the conflict to the kernel debug buffer (which is typically echoed to /var/log/messages). Before the error is logged, the device driver determines whether a SCSI Persistent Reservation is active on the target tape drive. If it is, it gets the reserving host initiator WWPN (worldwide port name). If successful, the device driver posts the message

```
lin_tape: reserve held by xxxxxxxx
```

to the debug buffer. To prevent multiple identical entries in the error log, subsequent reservation conflicts from the same reserving host WWPN are not logged.

Devices not reported at /proc/scsi/IBMchanger or /proc/scsi/IBMtape

After lin_tape is installed, following the [Installation and Configuration instructions](#), this command shows all tape devices and their device index:

```
cat /proc/scsi/IBM*
```

If any or some devices that are expected to show up are not there, confirm that they are attached to Linux SCSI layer by:

```
cat /proc/scsi/scsi
```

If the devices do not show up there, review your HBA configuration and device attachment. If they are there, make sure that they are supported devices at SSIC. See [Hardware requirements](#).

If files /proc/scsi/IBMchanger and /proc/scsi/IBMtape do not exist, review [Installation and Configuration instructions](#), and confirm lin_taped is running by using:

```
lin_taped status
```

and also that lin_tape is correctly installed on SLES and RHEL by using:

```
rpm -qa lin_tape
```

On Ubuntu:

```
apt list --installed | grep lin-tape
```

Windows Tape and Medium Changer device driver

This chapter describes the hardware requirements, software requirements, and installation notes for the Microsoft Windows device drivers for IBM® tape devices.

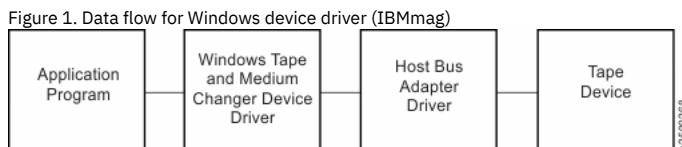
Purpose

The Windows tape and medium changer device driver is designed to take advantage of the features that are 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 unmount the cartridges), and 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 that is described here covers the Windows device driver and the interface between the application and the tape device.

[Figure 1](#) illustrates a typical data flow process.



- [Product requirements](#)
- [Installation and configuration instructions](#)
- [Persistent Naming Support on Windows Server 2008, 2012, 2016, and 2019](#)
- [Control Path failover support for tape libraries](#)
- [Data Path failover support for tape drives](#)
- [System-managed encryption](#)
- [Problem determination](#)
- [Old releases: V6.2.5.2 and prior releases](#)

Product requirements

Hardware requirements

Refer to the [Hardware requirements](#) for the latest hardware that is supported by the IBM® tape device driver.

Software requirements

For current software requirements, refer to the [Software requirements](#).

Note: Limited support for customers who have Microsoft Windows Server 2008 extended support from Microsoft only.

Installation and configuration instructions

This section includes instructions for installing and configuring the Windows tape and medium changer device driver on Windows Server 2008, 2012, 2016, and 2019.

The recommended procedure for installing a new version of the device driver is to uninstall the previous version (see [Uninstalling the device drivers](#)).

- [Windows Server 2008, 2012, 2016, and 2019 instructions](#)
- [Configuring limitations](#)

Windows Server 2008, 2012, 2016, and 2019 instructions

This section describes how to install, remove, and uninstall the Windows tape and medium changer device drivers on Windows Server 2008, 2012, 2016, and 2019.

Installation overview

The installation process consists of the following steps:

1. Verify that the hardware and software requirements are met.
2. Install the host bus adapters and drivers.
3. Shut down the system.
4. Connect the tape and medium changer devices to the host bus adapters.
5. Power® on the tape and medium changer devices.
6. Restart the system.
7. Log on as Administrator.
8. Install and configure the devices and device drivers with the installation application.

Installation procedures

These procedures make the following assumptions:

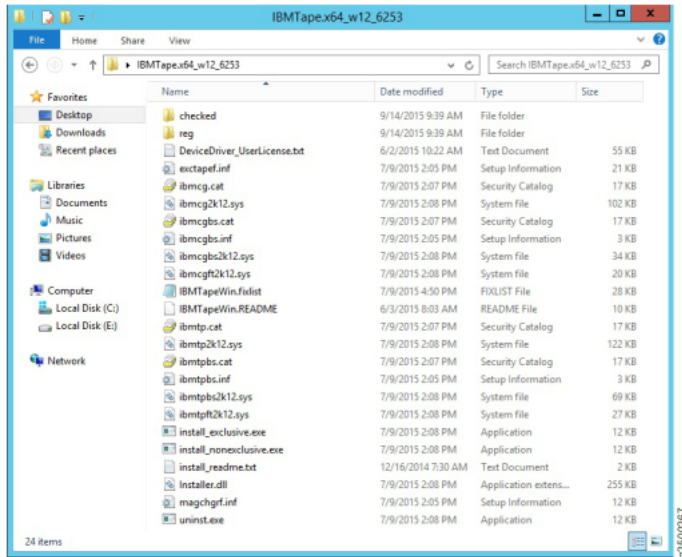
- No other driver is installed that claims the tape and medium changer devices.
- If you are updating the device driver from a Microsoft certified version to an uncertified version, you must first uninstall the certified driver. Refer to [Uninstalling the device drivers](#).
- The host bus adapter is installed, configured properly, and is running supported microcode and driver levels.
- Device drivers are installed with Windows Server 2012 or 2012R2, with V6.2.5.3 or later; Windows Server 2016, with V6.2.6.0 or later; or Windows Server 2019, with V6.2.6.8 or later.
Note: The latest driver level to include support for Windows Server 2008 is V6.2.5.2
- Different registry keys are created to configure different parameters. They can be at System\CurrentControlSet\Services\ and the subkeys are created depending on the Windows Server version. With Windows Server 2008 they are:
 - ibmcg2k8
 - ibmtp2k8
 - ibmtpbs2k8With Windows Server 2012 they are:
 - ibmcg2k12
 - ibmtp2k12
 - ibmtpbs2k12With Windows 2016, they are:
 - ibmcg2k16
 - ibmtp2k16
 - ibmtpbs2k16With Windows 2019, they are:
 - ibmcg2k19
 - ibmtp2k19
 - ibmtpbs2k19Refer to this list when in doubt of the registry key's name and instructions that involve modifying the registry. Caution and a backup are advised due to the registry's nature.
- 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 current one.
 - Windows Server 2019 for extended 64-bit architectures (Intel EM64T and AMD64),

IBMTape.x64_w19_nnnn.zip

To install the device drivers, follow this procedure.

1. Log on as Administrator.
2. Download the appropriate driver. Refer to [Accessing documentation and software online](#).
3. Extract the driver package to a hard disk drive directory of your choice, other than the root directory.
4. Ensure that the tape and medium changer devices are connected to your host bus adapter and configured properly by locating the devices in Device Manager.
5. Double-click either install_exclusive.exe or install_nonexclusive.exe.
 - install_exclusive.exe The driver issues automatic reserves on open. It also prevents multiple open handles from the host to a drive from existing at the same time, as is required by applications such as Tivoli® Storage Manager. This driver is also required for the failover feature to work as it uses persistent reservation (by default).

- `install_nonexclusive.exe` The driver permits open handles from the host to a drive to exist at the same time, as is required by applications such as Microsoft Data Protection Manager (DPM).



- a. More installation features are available through the command prompt interface (CLI), which include
- Installing only the tape or medium changer device drivers (-t or -c)
 - Running in debug mode, which creates the file debug.txt in the driver package directory (-d)
 - Running in silent mode, which suppresses messages that require user intervention, but only with Microsoft-certified IBM drivers (-s)
 - Disabling DPF from installation (-f), available in driver packages v6.2.0.1 and later
 - Enabling Persistent Reserve from installation if DPF is disabled (-p), available in driver packages v6.2.0.6 and later.
 - Disable Media Polling (-m).
 - Disable Dynamic Runtime Attributes (-a).
 - Exclude devices from being claimed by the driver (-e), available in driver packages v6.2.5.3 and later.

- b. If the Windows Found New Hardware Wizard begins during installation, cancel the wizard. The installation application completes the necessary steps.
6. If you are installing a Windows Server 2008, Windows Server 2012, 2016, or 2019 driver that is not certified by the Microsoft Windows Hardware Quality Laboratories (WHQL), it likely has a VeriSign digital signature. During installation, you might be presented with a prompt to install the software. Mark the Always trust software from IBM Corporation check box and click Install. You see this screen the first time that you install the drivers, provided you click the Always trust software box.

"What are the exact exceptions? Are cross-signed drivers still valid? Enforcement happens only on fresh installations, with Secure Boot on, and applies only to new kernel mode drivers:

To summarize, on non-upgraded fresh installations of Windows 10, version 1607 with Secure Boot ON, drivers must be signed by Microsoft or with an end-entity certificate issued before 29 July 2015 that chains to a supported cross-signed CA."

Go to https://blogs.msdn.microsoft.com/windows_hardware_certification/2016/07/26/driver-signing-changes-in-windows-10-version-1607/ for details.



Device removal or disable procedure

1. Right-click My Computer, select Manage to open the Computer Management Console, and click Device Manager.

2. Right-click the device that you want to uninstall and select Uninstall If you want to disable the device without uninstalling it, you can select Disable.
3. You are prompted to confirm the uninstallation. Click OK.
4. In Device Manager, under System devices, right-click Changer Bus Enumerator and select Uninstall.
5. In Device Manager, under System devices, right-click Tape Bus Enumerator and select Uninstall.

Note: This removal procedure removes the device from the device tree, but it does 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, complete the following steps.

1. Quiesce all activity on the tape and medium changer.
2. Double-click uninst.exe in the driver package.

Note: This action removes all the files in the system directories that were created during the installation of the device driver. It does not delete the compressed file or the files that were extracted from the compressed file. If you want to remove these files, you must delete them manually. For v.6.2.5.3 and later, it is not required to manually remove the devices at the Device Manager.

3. Restart the system.

Configuring limitations

The driver limitation for the supported number of tape devices is 1024. Every installed device uses a certain amount of resources. The user must also consider other resources, such as physical memory and virtual space on the system before you attempt to reach the limits. Also, be aware of Microsoft limitations. One known article is <http://support.microsoft.com/kb/310072> (ID: 310072). Be aware of any Windows specific version limitations.

Persistent Naming Support on Windows Server 2008, 2012, 2016, and 2019

The Windows tape driver has an option for enabling device object names that persist across restarts of the operating system. For example, if your tape drive has the name \\tape4801101 and the persistent naming option is used, then \\.\tape4801101 is reserved for use by that device after an operating system restart.

Complete the following steps to enable this feature.

1. Add a DWORD value to the registry called PersistentNaming and assign it a value 1
 - On Windows Server 2008: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtp2k8**
 - On Windows Server 2012: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtp2k12**
 - On Windows Server 2016: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtp2k16**
 - On Windows Server 2019: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtp2k19**
2. Restart your system. Then, the system writes information to the registry to associate the Worldwide Node Name from Inquiry p. 0x83 with the persistent name used by the operating system.
 - If the Worldwide Node Name is unavailable, or the drive is a virtual (that is, emulated) drive, then the device serial number is used rather than the Worldwide Node Name.
 - If the PersistentNaming option is not specified in the registry, then your devices might not be able to claim the same device name after restart or driver initialization.

You can find registry subkeys with persistent naming information.

- On Windows Server 2008: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtpbs2k8**
- On Windows Server 2012: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtpbs2k12**
- On Windows Server 2016: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtpbs2k16**
- On Windows Server 2019: **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\ibmtpbs2k19**

Alternately, you can use the Windows Device Manager to examine the device number to determine that persistent naming is enabled on your host. Persistent names contain tape device numbers that are based at 4801101 (which is the decimal equivalent of hexadecimal 0x49424D and ASCII "IBM®").

If two physical paths exist to a drive and different Windows device names are required (which happens, for example, when two different HBAs are connected to the drive and Data Path failover is disabled), the first discovered path claims the persistent device name. Any subsequent paths that connect to the same device receive names according to the order in which they are discovered by the Windows Device Manager.

Note: Persistent naming is not set by default. For disabling it, set the PersistentNaming value to 0 and restart the system.

Control Path failover support for tape libraries

To take advantage of Windows Control Path failover (CPF) support, the appropriate feature code must be installed. Refer to [Automatic failover](#) for what feature code might be required for your machine type.

- [Configuring and unconfiguring Control Path failover support](#)
- [Querying primary and alternative path configuration](#)
- [Checking disablement of Control Path failover setting](#)

Configuring and unconfiguring Control Path failover support

Control Path failover is enabled automatically when the device driver is installed by default (install_exclusive.exe). It can be disabled from installation with the -f CLI option. Or, it can be disabled or reenabled for the entire set of attached medium changers by modifying the registry.

1. Open the reg folder of the driver package.
2. Double-click DisableCPF.reg or EnableCPF.reg.
3. Reboot the system. This action is necessary for any registry modification to take effect.

Querying primary and alternative path configuration

To check whether the control path failover is enabled in the device driver and show the primary and alternative paths, use the tape diagnostic and utility tool.

Note: Show the primary and alternative path configuration for any device with tape diagnostic and utility functions. Refer to [IBM Tape Diagnostic Tool \(ITDT\)](#).

Checking disablement of Control Path failover setting

If you disabled the control path failover in the device driver's setting by double-clicking the DisableCPF.reg file and rebooting your system, you can go into the registry by issuing the Windows **regedit** command to confirm that CPF is disabled. Look for a line like

- On Windows Server 2008: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmcg2k8**
- On Windows Server 2012: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmcg2k12**
- On Windows Server 2016: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmcg2k16**
- On Windows Server 2019: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmcg2k19**

This line indicates that CPF is disabled in the driver. This setting takes effect only after your system is rebooted.

Data Path failover support for tape drives

To take advantage of Windows Data Path failover (DPF) support, the appropriate feature code must be installed. Refer to [Automatic failover](#) for what feature code might be required for your machine type.

- [Configuring and unconfiguring Data Path failover support](#)
- [Reserve Type if DPF is disabled](#)
- [Querying primary and alternative path configuration](#)
- [Checking disablement of Data Path failover setting](#)

Configuring and unconfiguring Data Path failover support

Data Path failover is enabled automatically when the device driver is installed by default (install_exclusive.exe). It can be disabled from installation with the -f CLI option. Or, it can be disabled or reenabled for the entire set of attached drives or medium changers by modifying the registry.

1. Open the reg folder of the driver package.
2. Double-click DisableDPF.reg or EnableDPF.reg.
3. Reboot the system. This action is necessary for any registry modification to take effect.

For latest LTO generation 3 code and later LTO generation tape drives, a license key feature for the library hardware is required.

Note: For LTO generation 3 or lower, or for tape drives that require a data path license key on the host side to enable DPF, the device driver looks for a file that is called %system_root%\IBM_DPF.txt for the key. %system_root% is the drive letter where Windows is installed, typically C (for example, C:\IBM_DPF.txt). The file contains the key on a single line, with no spaces and no other text on the line. If multiple keys are required, place each key in the file on its own line. The driver looks for this file at initialization. If the file contains a valid DPF license key, the DPF feature is enabled and any eligible devices have multi-path support.

Reserve Type if DPF is disabled

If DPF is disabled, SCSI-2 reserve is used by default to handle the reservation on tape drives. If Persistent Reserve is wanted rather than SCSI-2 reserve, ReserveTypePersistent.reg enables it (ReserveTypeRegular.reg disables it and then SCSI-2 reserve is used). Or, -p CLI option installation (only if -f was used to disable DPF) enables Persistent Reserve from installation.

Note: If DPF is not disabled, Persistent Reserve is used.

Querying primary and alternative path configuration

To check whether the data path failover is enabled in the device driver and show the primary and alternative paths, you can use the tape diagnostic and utility tool.

Note: Show the primary and alternative path configuration for any device with the tape diagnostic and utility functions. Refer to [IBM Tape Diagnostic Tool \(ITDT\)](#).

Checking disablement of Data Path failover setting

If you disabled the data path failover in device driver's setting by double-clicking the DisableDPF.reg file and rebooting your system, you can go into the registry by issuing the Windows **regedit** command to confirm that DPF is disabled. Look for a line

- On Windows Server 2008: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k8**
- On Windows Server 2012: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k12**
- On Windows Server 2016: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k16**
- On Windows Server 2019: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k19**

This line indicates that DPF is disabled in the driver. This setting takes effect only after your system is rebooted.

If you enabled Persistent Reserve on DPF disabled, look for a line

- On Windows Server 2008: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k8**
- On Windows Server 2012: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k12**
- On Windows Server 2016: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k16**
- On Windows Server 2019: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k19**

If you use ReserveTypeRegular.reg, then the **ReserveType** value is set to 0.

System-managed encryption

- [Device driver configuration](#)
- [Configuration file](#)
- [Querying tape drive configuration settings](#)

Device driver configuration

System-managed encryption parameters on Windows are placed in the registry under the key for the device driver. The parameters are populated in user-created subkey that contains the serial number of the device. The registry keys (sys_encryption_proxy and sys_encryption_write) are used to determine SME enablement and invocation of the EKM proxy on write.

Note: Leading zeros in the serial number must be excluded. For example, if the serial number of the encryption-capable tape drive were 0123456789, the user creates the following registry:

- On Windows Server 2008: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k8**
- On Windows Server 2012: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k12**
- On Windows Server 2016: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k16**
- On Windows Server 2019: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k19**

Under this key, the user creates DWORD values that are called sys_encryption_proxy or sys_encryption_write, and assigns them values that correspond with the wanted behavior.

The device driver SME settings can be set for all drives at once by placing the sys_encryption_proxy and sys_encryption_write registry options under the device driver key, found at:

- On Windows Server 2008: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k8**
- On Windows Server 2012: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k12**
- On Windows Server 2016: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k16**
- On Windows Server 2019: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k19**

When this option is chosen, the settings that are established for all drives are overridden by the serial-number specific settings described the previous paragraph.

If no options are specified in the registry, the driver uses the default values for the parameters.

- The default value for sys_encryption_proxy is 1.
This value causes the device driver to handle encryption key requests, if the drive is set up for system-managed encryption. This value does not need to be changed. A value of 0 causes the device driver to ignore encryption key requests for system-managed encryption drives, and is not desirable.
- The default value for sys_encryption_write is 2.
This value causes the device driver to leave the encryption write-from-BOP settings alone. It does not turn on or turn off encryption writing, but instead uses the settings that are already in the drive. If encryption is not set up previously, then the drive writes unencrypted data. A value of 0 causes the device driver to write unencrypted data. A value of 1 causes the device driver to write encrypted data.

Changes to the registry require a reboot before the settings are able to be viewed. However, during new installations of the driver, if the old driver is not uninstalled, the old settings remain in place and no reboot is required.

Configuration file

The file %system_root%\IBMEKM.conf is used to store the IP address of the EKM server and other network-related parameters. The phrase %system_root% refers to the drive letter where the Windows installation is located, typically C (for example C:\IBMEKM.conf).

The format for the EKM server parameters is

Server<tab>Timeout<tab>IPAddress:Port

For example, if the Windows installation is installed at c:\, the EKM server is named ekmttest, the wanted timeout is 10 seconds, and the IP address is 127.0.0.1 at port 4242. Then, create a line in C:\IBMEKM.conf as follows.

```
ekmttest<tab>10<tab>127.0.0.1:4242
```

Querying tape drive configuration settings

An ITDT command is used to query the encryption settings of a tape drive. In ITDT-SE this procedure can be done by using either the [Standard Edition - Scan menu commands](#) section or the Tapeutil menu commands with command [\[35\] Query Encryption Status](#).

In ITDT-GE, it can be accomplished by using the Diagnostic/Maintenance tab (see [Encryption](#)) or by [Query Encryption Status](#) on the Tapeutil tab.

Problem determination

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 Debug View from Sysinternals Suite, available from <http://technet.microsoft.com/sysinternals/>.

- [Windows Server 2008, 2012, 2016, and 2019 instructions](#)
- [Reservation conflict logging](#)
- [Max retry busy](#)
- [Checking if Data Path failover is enabled](#)
- [Checking if Control Path failover is enabled](#)

Windows Server 2008, 2012, 2016, and 2019 instructions

Windows 2008, 2012, 2016, and 2019 instructions

1. Quiesce all activity on the tape and medium changer.
2. Exit all applications that are using the tape and medium changer devices.
3. If the driver is installed, complete the Uninstall the devices procedure that is described in a previous section.
4. Locate the \checked folder for the device driver level that you are running. This folder is in the highest level directory of the driver package. It contains checked versions of the tape and medium changer device drivers, ibmtpxxyy.sys, and ibmcgxyyy.sys, where
 - xx = ft for the filter driver, bs for the bus driver, or blank for the base driver
 - yyyy = 2k8. It also has the .inf and .cat files on Windows Server 2008.
 - yyyy = 2k12. It also has the .inf and .cat files on Windows Server 2012.
 - yyyy = 2k16. It also has the .inf and .cat files on Windows Server 2016.
 - yyyy = 2k19. It also has the .inf and .cat files on Windows Server 2019.
5. Copy all the files from the \checked folder to the highest level directory of the driver package, overwriting the files in there. A previous backup of the files or having a copy of the original driver package is recommended.
6. Start Dbgview.exe to capture debug messages. Dbgview can be downloaded from the Microsoft website. Make sure that Capture Kernel, Enable Verbose Kernel Output, and Capture Events are checked at the Capture menu.
7. Complete the Installation procedure.
8. Issue commands to the driver. You see debug messages on Dbgview from IBMTpBus, IBMCGBus, tape, or mcd. For example, you might see IBMTpBus: ENT: tag output. If you do not see something similar, an error might have happened during the checked driver installation or no driver activity might have occurred.

Restoring the non-Debug version

To restore the non-debug version of the driver, complete the following steps.

1. Quiesce all activity on the tape and medium changer devices.
2. Exit all applications that are using the tape and medium changer devices.
3. Uninstall the debug driver version, which can be accomplished by running uninst.exe, within the driver's package folder.
Note: For v.6.2.5.3 and later, there is no need to manually remove the devices at Device Manager.
4. Restart the system.
5. When the system is back, install the non-debug driver version.

Reservation conflict logging

When the device driver receives a reservation conflict during open or after the device is opened, it logs a reservation conflict in the Windows eventlog. Before the error is logged, the device driver issues a **Persistent Reserve In** command. This action determines whether a SCSI Persistent Reservation is active on the reserving host to get the reserving host initiator WWPN (worldwide port name). If successful, the device driver logs this information as follows.

```
Reserving host key: kkkkkkkk WWPN: xxxxxxxx
```

Where kkkkkkkk is the actual reserve key and xxxxxxxx is the reserving host initiator WWPN.

After initially logging the reserving host WWPN, subsequent reservation conflicts from the same reserving host WWPN are not logged. This action prevents multiple entries in the error log until the reserving host WWPN is different from the one initially logged. Or, the device driver reserved the device and then another reservation conflict occurs.

Max retry busy

Complete the following steps to enable this feature.

1. Add a DWORD value to the registry called MAXBusyRetry and assign it a value between 1 and 480 at:
 - On Windows Server 2008: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k8`
 - On Windows Server 2012: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k12`
 - On Windows Server 2016: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k16`
 - On Windows Server 2019: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmtp2k19`
2. Reboot your system. Then, when a device gets the device busy error, the command is retried up to *MAXBusyRetry* times.

Checking if Data Path failover is enabled

If the tape diagnostic utility is not available, you can confirm that Data Path failover is enabled by installing the Debug version of the device driver. Refer to [Problem determination](#) or [Old releases: V6.2.5.2 and prior releases](#). While the Debug device driver is installing, look for the message: `tape.CreateTapeDeviceObject: DPF INSTALLED`.

Checking if Control Path failover is enabled

If the tape diagnostic utility is not available, you can confirm that Control Path failover is enabled by installing the Debug version of the device driver. Refer to [Problem determination](#) or [Old releases: V6.2.5.2 and prior releases](#). While the Debug device driver is installing, look for the message: `FtlValidateCpfKey` and confirm that the status is `True`.

Old releases: V6.2.5.2 and prior releases

Installing the Device Drivers

The last driver level to include support for Windows 2000 is V6.1.4.8.

The last driver level to include support for Windows 2003 is V6.2.1.7.

The last driver level to include support for Windows 2008 is V6.2.5.2.

Subsequent levels include support for Windows Server 2012; support for Windows Server 2016, starting with level 6.2.6.0; and support for Windows Server 2019, starting with level 6.2.6.8.

Windows Server 2008 - 2019 instructions

1. Refer to [Installation and configuration instructions](#). The same installation instructions for Windows Server 2019 apply to Windows Server 2012 and Windows Server 2008.

Uninstalling the Device Drivers

1. Complete the steps under [Device removal or disable procedure](#) to remove the tape and medium changer devices.
2. Double-click `uninst.exe` in the driver package.
Note: This action removes all the files in the system directories that were created during the installation of the device driver. It does not delete the compressed file or the files that were extracted from the compressed file. If you want to remove these files, you must delete them manually.
3. Restart the system.

Installing the Debug version

Windows Server 2008 - 2019 instructions

1. Quiesce all activity on the tape and medium changer.
2. Exit all applications that are using the tape and medium changer devices.
3. If device drivers are installed, uninstall the device drivers first. Refer to [Uninstalling the Device Drivers](#).
4. Locate the `\checked` folder in the driver package.
5. Copy the files from the `\checked` folder to the highest level directory of the driver package, overwriting the files there. A previous backup of the files or having a copy of the original driver package is recommended.
6. Complete the Installation procedure.

Restoring the non-Debug version

To restore the non-debug version of the driver, complete the following steps.

Windows Server 2008 - 2019 instructions

1. Quiesce all activity on the tape and medium changer.
2. Exit all applications that are using the tape and medium changer devices.

3. Uninstall the device drivers first. Refer to [Uninstalling the Device Drivers](#).
4. Restart the system.
5. If a backup of device drivers package was done, complete the installation by using backup. Otherwise, download the same device driver level again and complete the installation. When the driver starts and commands are issued to it, the driver no longer produces debug output.

IBM Tape Diagnostic Tool (ITDT)

This chapter describes the IBM® Tape Diagnostic Tool.

- [Purpose](#)
- [Accessing ITDT](#)
- [Supported systems](#)
- [IBM Tape Diagnostic Tool - Standard Edition](#)
- [IBM Tape Diagnostic Tool - Graphical Edition](#)

Purpose

The IBM® Tape Diagnostic Tool (ITDT) is available in two versions:

- Standard Edition (ITDT-SE) - The command line version.
- Graphical Edition (ITDT-GE) - The GUI version for the following platforms:
 - Microsoft Windows operating systems
 - Linux® operating systems

Both versions provide the user with a single diagnostic program for tapeutil applications. Both SE and GE contain tapeutil functions with SE also providing scripting capability.

Note: The term *tapeutil* is a synonym for the tool that is delivered with the device driver. For example, this tool is named tapeutil on UNIX operating systems; it is named ntutil on Microsoft Windows operating systems.

ITDT supports IBM tape and library devices claimed by the IBM Tape Device Driver and devices that use the generic Operating System driver. By default the IBM Tape Device Driver is used but only if the IBM Tape Device Driver is not installed or the device is not claimed by the IBM Tape Device Driver. The generic Operating System driver is used when available.

The available advanced operations that are provided by the IBM Tape Diagnostic Tool are completed on tape drives and tape libraries. By using this function, you can complete maintenance tasks and run diagnostic tasks to determine tape drive issues. This action reduces product downtime and increases productivity.

The IBM Tape Diagnostic Tool is designed to

- Run quick or extended diagnostic tests on tape drives.
- Start tape library self-test operations.
- Retrieve dumps from tape drives and libraries.
- Run a firmware update on tape drives or libraries.
- Online Check for Firmware updates for tape drives or libraries (ITDT-GE).
- Test the performance of the environment by completely writing a cartridge and measuring performance.
- Verify tape drive compression.
- Measure system performance.
- Retrieve and display cartridge usage information.
- Verify the encryption environment.

This test is used to verify whether data on the cartridge was written encrypted.
- Scan the system to discover all supported tape and library devices.
- Run a connection test (scan).

This test is used to verify that all devices are attached properly.
- Run a standard test to check whether the tape device is defective and output a pass/fail result.

Note: When this test is completed, all data on the cartridge is overwritten.
- Run a full write function.

This function writes the entire cartridge, overwriting all previous data with a selectable block size that contains either compressible or incompressible data and then outputs performance data.

Note: When this test is completed, all data on the cartridge is overwritten.
- Run a system test.

Write different block sizes with compressible and incompressible data and then outputs performance data.

Note: When this test is completed, all data on the cartridge is overwritten.
- Run a tape usage function to retrieve statistical data and error counters.
- Run HD-P functions like discovery.
- Physical Copy/Data migration and verification of cartridges.
- Log and Dump File analysis.

ITDT-SE provides the most important functions of the previous tapeutil tools. As an extension of the current tapeutil variants, the set of operations and functions available with ITDT-SE is identical across all supported operating systems (unless a function is not available on a particular system).

Dedicated device drivers for tapes and libraries can be installed on the target system and an application is installed that uses the tape/library devices. When this configuration exists, ITDT-SE can coexist with the application so that when the application disables the device internally, ITDT-SE can run the diagnostic tests on that device.

Accessing ITDT

IBM maintains the latest levels of the ITDT tool chain and documentation on the Internet at <http://www.ibm.com/support/fixcentral>.

One option to access ITDT is through [Accessing documentation and software online](#). When an IBM driver is downloaded, there is a corequisite to download ITDT.

This portal gives access to the download area where the following procedure guides you to the correct download:

1. In the Product Group menu, select Storage Systems.
2. In the Product Family menu, select Tape Systems.
3. In the Product Type menu, select Tape device drivers and software.
4. In the Product menu, select Tape Diagnostic Tool (ITDT).
5. Select your platform and press Continue.

Supported systems

ITDT is developed to support various versions of different platforms and tape products. For the latest support, refer to [Introduction](#) and Inter-operation Center website product support at <http://www.ibm.com/systems/support/storage/config/ssic/>.

IBM® Tape Diagnostic Tool - Standard Edition

- [Installing ITDT - Standard Edition](#)
- [Starting ITDT - Standard Edition](#)
- [Generic Operating System driver with ITDT-SE](#)
- [Standard Edition - known issues and limitations](#)
- [Standard Edition - Start menu commands](#)
- [Standard Edition - Scan menu commands](#)
- [Standard Edition - Tapeutil menu commands](#)
- [Add device manually](#)
- [Standard Edition - Program options](#)
- [Standard Edition - Tapeutil scripting commands](#)

Installing ITDT - Standard Edition

This section describes the installation procedures for the Standard Edition of ITDT in various operating systems.

Before the IBM® Tape Diagnostic Tool Standard Edition (ITDT-SE) is used with the IBM Tape Device Driver, we recommend upgrading to the latest available IBM Tape Device Driver level.

In a System Managed Encryption setup, the Encryption test [E] always exits with NO DRIVER SPECIAL FILE when ITDT-SE is started with **-force-generic-dd**.

- [Installing ITDT-SE on AIX operating systems](#)
- [Installing ITDT-SE on Microsoft Windows operating systems](#)
- [Installing ITDT-SE on i5/OS operating systems](#)
- [Installing ITDT-SE on other supported operating systems](#)

Installing ITDT-SE on AIX operating systems

ITDT-SE on AIX® operating systems has two installation modes and can be installed by one of the following methods.

1. Installation in the AIX software repository:
 - a. Download `tape.itdt.<version>.bff`
 - b. Change to the download folder and run the **inutoc** command.
 - c. To install the file, run the following command.

```
installp -acd tape.itdt.<version>.bff all
```

Or

```
smitty install
```
2. Installation independently.
 - a. Download `install_itdt_se_<OS>_<version>` to a directory of your choice.
 - `install_itdt_se_Aix_<version>` is for AIX® operating systems.
 - b. Run the following command to make `install_itdt_se_<OS>_<version>` executable.

```
chmod 700 install_itdt_se_Aix_<version>
```

3. To run without parameters, run the following command.

```
install_itdt_se_Aix_<version>
```

Or

```
./install_itdt_se_Aix_<version>
```

depending on your operating system.

Note: ITDT-SE can be used only by a user with root access rights.

Installing ITDT-SE on Microsoft Windows operating systems

To install ITDT-SE on Windows operating systems, complete the following steps.

Note: ITDT-SE can be used only by a user with administrator rights.

1. Download `install_itdt_se_WindowsX86_64_<version>.exe` to a directory of your choice.
2. Run the following command:

```
install_itdt_se_WindowsX86_64_<version>.exe
```

Installing ITDT-SE on i5/OS operating systems

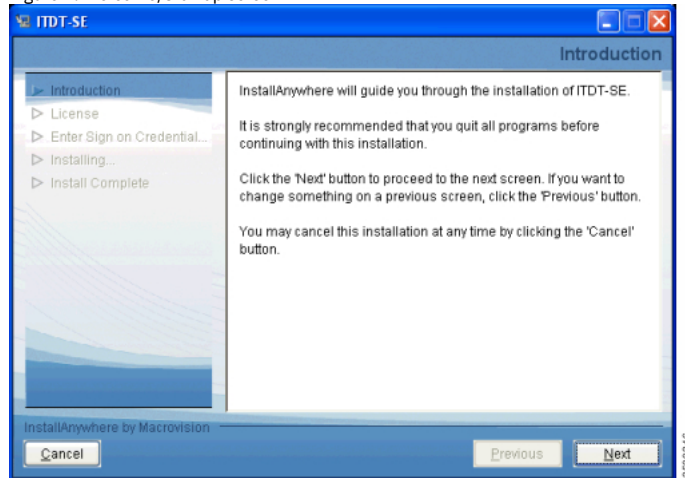
ITDT-SE for i5/OS or IBM i is installed with a graphical installer based on the InstallAnywhere installation software. This software must be run on a Microsoft Windows system and installs the software onto the remote i5/OS system. The installer generates installation information on the local Microsoft Windows system. So, the program can be removed from the remote i5/OS system by running the uninstaller on the local Microsoft Windows system.

To install ITDT-SE on i5/OS operating systems, complete the following steps.

Note: ITDT-SE on i5/OS requires a Microsoft Windows PC with a Java™ run time 1.4.2 or higher.

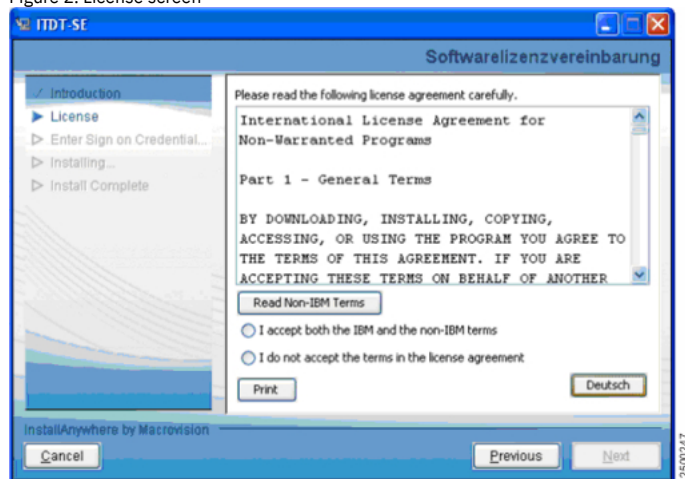
1. Start the installer `installi5OS_ITDT_SE_<version>.exe`.
2. Read through the welcome and license screens.

Figure 1. Welcome/Startup screen



3. Accept the License Agreement, then click Next.

Figure 2. License screen



4. Enter your i5/OS system host name or IP address and administrator credentials. Start the installation by pressing Install. Click Done when the installation is complete.

Note: The user name must be QSECOFR.

Figure 3. Machine credentials screen

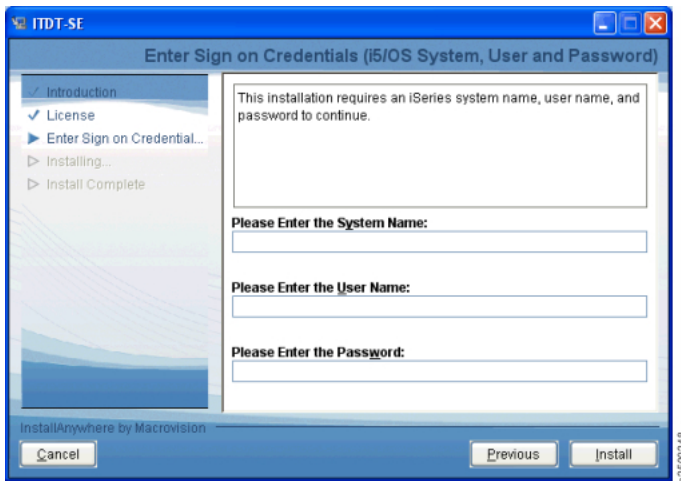


Figure 4. Installation progress screen

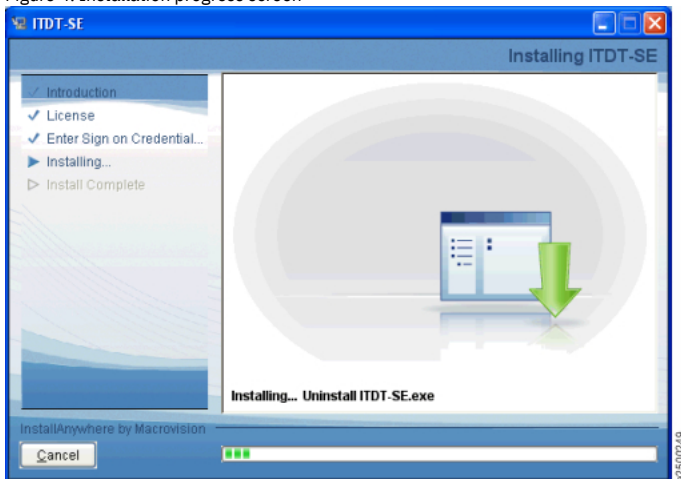
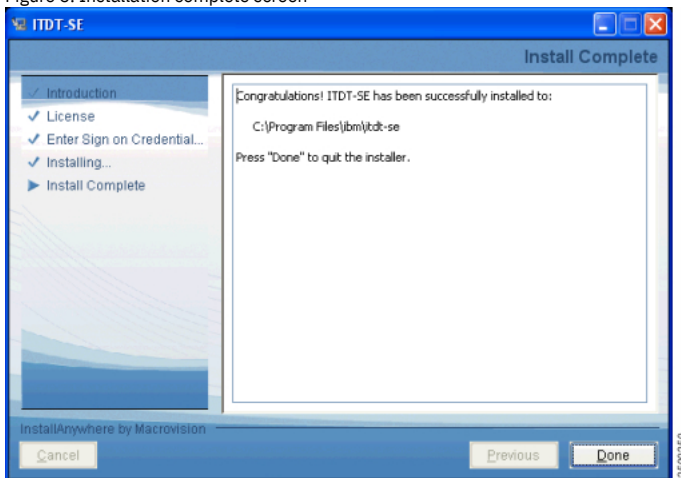


Figure 5. Installation complete screen



Installing ITDT-SE on other supported operating systems

To install ITDT-SE on other supported operating systems, complete the following steps.

Note: ITDT-SE can be used only by a user with root access rights, except for the Mac OS, which requires the user to have the minimum of read/write access to the device file.

1. Download install_itdt_se_<OS>_<version> to a directory of your choice.
 - install_itdt_se_Aix_<version> is for AIX® operating systems
 - install_itdt_se_Linuxx86_64_<version> is for Linux® operating systems on X86_64 hardware
 - install_itdt_se_Linuxpowerpc64_<version> is for Linux operating systems on pSeries
 - install_itdt_se_Linuxpowerpc64le_<version> is for Linux operating systems on pSeries Little Endian.
 - install_itdt_se_Linuxs390x_<version> is for Linux operating systems on zSeries
 - install_itdt_se_MacOS_<version> is for Mac OS operating systems.

2. Run the following command to make `install_itdt_se_<OS>_<version>` executable.

```
chmod 700 install_itdt_se_<OS>_<version>
```

3. To run without parameters, run the following command

```
install_itdt_se_<OS>_<version>
```

or

```
./install_itdt_se_<OS>_<version>
```

depending on your operating system.

Optionally you can use these parameters:

- `-h [--help]` Show help message
- `-v [--verbose]` Print the extracted files
- `-p [--path]` Path for extraction (default is current directory)
- `-t [--test]` Show package content

Starting ITDT - Standard Edition

This section describes the startup procedures for the Standard Edition of ITDT in various operating systems.

- [Starting ITDT-SE on Solaris operating systems](#)
- [Starting ITDT-SE on Windows operating systems](#)
- [Starting ITDT-SE on i5/OS operating systems](#)
- [Starting ITDT-SE on other supported operating systems](#)

Starting ITDT-SE on Solaris operating systems

1. If the IBM® Tape Device Driver is not used, if it is not installed, or you want to be able to run **-force-generic-dd**, configure the *sgen* driver. A script `sgen_solaris_conf.sh` is included in the ITDT-SE package. This script allows you to configure the *sgen* generic SCSI driver that is shipped with Solaris.

ITDT-SE requires that the *sgen* device driver is configured so that the devices you want to work with can be found.

Note: For system security reasons, always reset the *sgen* device driver settings after you finish working with ITDT-SE, by using the `sgen_solaris_conf.sh` script. To configure the *sgen* driver, start the `sgen_solaris_conf.sh` script with root access.

The following command-line options are available on the *SGEN* driver configuration screen:

- 1. Check driver: This option checks if the driver `/kernel/drv/sgen` is available.
- 2. List driver settings: This option shows the current activated devices.
- 3. New driver configuration: This option shows the screen that is used to create a new driver configuration (see Step 2).
- 4. Stop *sgen* driver: This option stops the driver (that is, `rem_drv sgen`).
- 5. Start *sgen* driver: This option stops and starts the *sgen* driver.
- 6. Exit program: This option closes the shell script.

2. Enter option 3 to create a new driver configuration.

The following command-line options are available on the New Configuration screen. Use these options to configure the *sgen* driver:

- 1. List targets: This option shows the targets in current configuration.
 - 2. Define device types: This option defines drive and changer types.
 - 3. or 5. Add targets: This option adds targets to the list.
Note: Option 3 allows for the addition of individual devices one at a time. Option 5 allows for the addition of a range of devices, eliminating the need to add many devices one by one.
 - 4. or 6. Remove targets: This option removes targets from the list.
Note: Option 4 allows for the removal of individual devices one at a time. Option 6 allows for the removal of a range of devices, eliminating the need to remove many devices one by one.
 - 7. Save configuration: This option saves the modifications.
 - 8. Back to Main Menu: This option returns to the main menu.
3. After the *sgen* driver is configured, enter command-line option 8 to go back to the main menu.
4. On the *SGEN* driver configuration screen, enter command-line option 5. This option starts the *sgen* driver. New devices are found by using the definitions that are completed in Step 2.
5. After the new devices are found, enter option 6 to exit the `sgen_solaris_conf.sh` script.

Note: For Fibre Channel Host Bus Adapters (HBAs), special handling is required. Attached devices must be configured with their WWPN in the `sgen.conf` file. This task must be done manually. It is not completed by using the `sgen_solaris_conf.sh` script.

The following code is an example how to add those devices:

```
Run the command "cfgadm -al" to get the WWPN number(s).
.....
c4                                fc-private  connected  configured
c4::5005076302401924              tape        connected  configured
.....
Add the WW-PN number(s) into the sgen.conf file.
name="sgen" parent="fp" target=0 lun=0 fc-port-wwn="5005076302401924";
name="sgen" parent="fp" target=0 lun=1 fc-port-wwn="5005076302401924";

If you have finished the editing, the sgen driver has to be restarted.
Please enter "update_drv sgen".
```

The **Read Attribute** command is 16-byte CDB. On Solaris 10, IBMtape can detect what maximum CDB length is supported from HBA attributes and set the supported CDB. For Solaris 9, CDB16 must be enabled for ITDT-SE to work correctly. This procedure can be done by adding the entry of

```
cdb16_support=1
```

at the first line in

```
/usr/kernel/drv/IBMtape.conf
```

and reload the IBMtape driver again.

To start ITDT-SE, run the following command:

```
./itdt
```

At first start, read the User License Agreement.

- Press Enter to scroll the license screens forward or **b** followed by Enter to go back.
- Type **i** if you agree to the terms of license or **q** followed by Enter to quit the application.

During the initial program start, the input and output directories are created.

- Input directory: default directory for firmware files during Firmware Update
- Output directory: directory that contains the result files, dump files, and log files after tests are run

ITDT-SE does not make changes outside the installation directory.

Starting ITDT-SE on Windows operating systems

At first program start, the license text is displayed and the input and output directories are created. Start ITDT by running the following command from the directory "ITDT" that was created during the installation (confirm to run with administrator rights):

```
itdt.exe
```

Read the User License Agreement.

- Press Enter to scroll the license screens forward or **b** followed by Enter to go back.
- Type **i**, if you agree to the terms of license or **q** followed by Enter to quit the application.

ITDT-SE does not create any registry entries or make changes outside the installation directory.

During installation a subdirectory "ITDT" was created which contains the ITDT program file and 2 subdirectories initially:

- "License" directory with license files in different languages.
- "Scripts" directory for extra scripts.

When the program is run for the first time, two more subdirectories are created in the "ITDT" folder:

- "Input" directory for firmware files to be uploaded to devices.
- "Output" directory for generated log and dump files.

To remove ITDT from your system, erase the ITDT directory. Any log and dump files in the subdirectories are also erased when you do so.

Starting ITDT-SE on i5/OS operating systems

To use ITDT-SE to update firmware or pull dumps from a tape drive inside a tape library, make sure that the drives are varied online in STANDALONE MODE by completing the following steps.

1. Issue the command **WRKMLBSTS**. Identify the library and drives you want to work with. Note their names (for example, TAPMLB01, TAP01, TAP02).
2. Deallocate the corresponding drives by using option 6.
3. Vary OFF the TAPMLB by using option 2.
4. Enter the following command:

```
WRKCFGSTS *DEV TAP*
```

Identify the drives that were noted in Step 1 (for example, TAPMLB01, TAP01, TAP02) and vary them ON by using option 1.

5. Start the iSeries Qshell environment with the following command:

```
QSH
```

6. Change to the folder /home/ITDT with the following command:

```
cd /home/ITDT
```

7. Start ITDT with the following command.

```
./itdt
```

8. Update Firmware and pull dumps. See [Firmware Update](#) and [Dump](#).
9. When firmware updates and dumps are complete, enter the following command:

```
WRKCFGSTS *DEV TAP*
```

10. Vary off the TAPs that you worked with by using option 2.

11. Issue the command **WRKMLBSTS**. Identify the library and drives you worked with.
12. Vary on the TAPMLB by using option 1.
13. Press F5 to refresh the screen. The TAPs belonging to the TAPMLB shows up.
14. Allocate the TAPs back to the TAPMLB by using option 4 or 5.

Starting ITDT-SE on other supported operating systems

To start ITDT-SE, run the following command

```
./itdt
```

At first start, read the User License Agreement:

- Press Enter to scroll the license screens forward or **b** followed by Enter to go back.
- Type **i**, if you agree to the terms of license or **q** followed by Enter to quit the application.

During the initial program start, the input and output directories are created.

- Input directory: default directory for firmware files during Firmware Update
- Output directory: directory that contains the result files, dump files, and log files after tests are run

ITDT-SE does not make changes outside the installation directory.

Generic Operating System driver with ITDT-SE

When the IBM Tape Device Driver cannot be used, ITDT-SE can be forced to use generic Operating System driver devices with the parameter **-force-generic-dd**.

The parameter is supported in the interactive and scripting mode of ITDT-SE.

For information, go to:

- [Standard Edition - Program options](#)
- [Device file names - device addressing](#)
- [Scan](#)

Standard Edition - known issues and limitations

This section describes the known issues and limitations of the ITDT-SE program.

- [AIX operating systems](#)
- [HP-UX operating systems](#)
- [Linux operating systems](#)
- [Solaris operating systems](#)
- [Windows operating systems](#)
- [i5/OS operating systems](#)
- [All supported operating systems](#)

AIX® operating systems

The following are the known scan limitations:

- Only devices that have the device state "available".

For FC and SAS devices, ID and LUN greater than 999 are not displayed; they are shown as ###.

When logged in through telnet, backspace might not work - an escape sequence is inserted and the input is ignored after Enter is pressed.

HP-UX operating systems

Verify that the following patches are installed before ITDT-SE is started.

- PA-Risc: At least these patches:
 - ld and linker tools cumulative patch
 - libc cumulative patch
- Itanium/IA-64: All regular patches and the following patches:
 - VxVM 3.5~IA.014 Command Patch
 - VxVM 3.5~IA.014 Kernel Cumulative Patch
 - Aries cumulative patch
 - linker + fdp cumulative patch

Note: ITDT-SE is emulated by Aries (a binary emulator that transparently emulates 32-bit and 64-bit HP-UX PA-RISC applications on HP-UX IA-64 machines). On HP-UX11.11 systems, tape libraries that are operated through the drive's control path (no control path failover) might disappear from the Device List after a [F] Firmware Update on the controlling drive. It is recommended to complete repeated [S] Scan operations to make the library reappear in the device list.

Linux operating systems

ITDT-SE on Linux® requires **glibc** 2.2.5 or later.

Note: On an SLES9 s390x (64 bit) configuration, you might experience a SCSI CMD TIMEOUT when the [T] option is run with the IBM® Tape Device Driver. For SUSE SLES9 on zSeries, ensure that the kernel update **SUSE-SA:2007:035** is installed.

Solaris operating systems

Rescan might take 6-8 minutes, depending on the numbers of host adapters and devices attached.

The known scan limitations: SCSI ID 0-255, LUN 0-10.

If the IBM® Tape Device Driver is not installed on Solaris 10, tape devices might not be found during scan although they are configured in **sgen.conf**. When this event occurs, complete the following steps to configure the devices:

1. Check the current driver bindings for IBM tape drives and changers by entering the following commands:

```
# egrep "scsiclass,01" /etc/driver_aliases      (for drives)
# egrep "scsiclass,08" /etc/driver_aliases      (for changers)
```

2. Modify the **/etc/driver_alias** file to comment all lines not starting with sgen and containing identification of your drives and changers. Examples:

```
#st "scsiclass,01"          (all tape drives)
#st "scsiclass,01.vIBM.pULT3580-TD4"      (IBM tape drive model ULT3580-TD4)
#st "scsiclass,08"          (all changers)
#st "scsiclass,08.vIBM.p3573-TL"      (IBM changer model 3573-TL)
```

3. Check that the configured drives are not configured for st driver by entering the following command:

```
# cfgadm -al
```

If the tape drive is claimed by st device driver, an entry with **cxx:rmt/y**, is displayed, for example:

```
c11::rmt/0      tape      connected      configured      unknown
```

4. Add sgen driver aliases with one of the following commands:

```
# update_drv -a -i '"scsiclass,01.vIBM.pULT3580-HH4"' sgen
  (adds sgen alias for IBM drive, model ULT3580-HH4)

# update_drv -a -i '"scsiclass,01"' sgen
  (adds sgen alias for all drives attached to the system)

# update_drv -a -i '"scsiclass,08.vIBM.pULT3581-TA2"' sgen
  (adds sgen alias for IBM changer, model ULT3581-TA2)

# update_drv -a -i '"scsiclass,08"' sgen
  (adds sgen alias for all changers attached to the system)
```

5. Check that the drives and changers are now configured with the following command:

```
# cfgadm -al
```

6. If the drives or changers are not listed in the output of '**cfgadm -al**', reboot the system and verify the list of configured devices with the command:

```
# cfgadm -al
```

Windows operating systems

After a firmware update, devices might disappear. This issue is a known Windows problem.

Repeated Scan operations can help to rediscover the device.

When applications are turned on Windows while ITDT-SE is running, an extra ESC character might appear on the input line. When this issue occurs, the input is ignored after Enter is pressed.

If you are using Adaptec SCSI Host Bus adapters, ensure that you are using the latest Adaptec Host Bus Adapter Drivers instead of the drivers that are shipped with the Windows operating system.

On Microsoft Windows systems where the maximum transfer size is limited to less than 64 kB, the Dump and Firmware Update operations do not work.

i5/OS operating systems

The Tape Drive must be varied online. If the tape drive is operated through a tape library, the library must be varied offline. See [Starting ITDT-SE on i5/OS operating systems](#) for details.

As the library is varied offline, the Encryption Test does not deliver decrypted data in a Library Managed Encryption environment.

IO adapters without a dedicated IOP do not support commands with 16 bytes. These adapters are called IOP-less and can be identified with the command; **WRKHDWRSC *STG**. Adapters where the CMBxxx and DCxxx resource names are the same (like both 5774) are IOP-less. If it is IOP, then the CMBxxx is a different type from the DCxxx.

Therefore, the following ITDT commands and test sequences can fail on IOP-less adapters attached devices.

- Encryption Verification Test
- Physical Copy with more than one partition
- **Erase** command
- **Change Partition** command
- **Read/write -attribute** command

ITDT-SE on i5/OS V5R4 requires the following PTF installed.

- PTF: SI25023 Release: 540 Abstract: OSP-MEDIA-TAPE-THREADS-MSGCEE0200-T/QTAHRMGR QTARDCAP FAILS

The [U] Tapeutil option is not available for i5/OS with this release as all the underlying operations require the IBM® Tape Device Driver to be installed.

FC 5912 SAS HBA support is only for POWER6 and V6R1 configurations that are attached to LTO Gen 4 HH tape drives (No support for LTO 3 HH SAS).

All supported operating systems

This section describes the known issues and limitations of the ITDT-SE program on all other supported operating systems.

Prevent/Allow Medium Removal is missing as a Tape Drive option. But, it can still be completed by using the [56] Prevent/Allow Medium Removal option for tape libraries while the tape device is opened.

User Interface issues

If you press the arrow keys on most UNIX operating system consoles, the input is ignored after Enter is pressed.

When the Tab key is pressed as an input string for field data, the user interface is corrupted.

Make sure that field input does not start with a number followed by space and extra text. This input is interpreted as an entry to a specific row in the field. To avoid this issue, use an underscore character (_) instead of the space character.

Command timeout

There is no instant operation termination upon SCSI command timeout; for example, when the SCSI cable is unplugged after POST A is started.

When a command timeout condition occurs, ITDT might still continue to complete more operations (like unmounting the cartridge) instead of instantly terminating with a timeout condition.

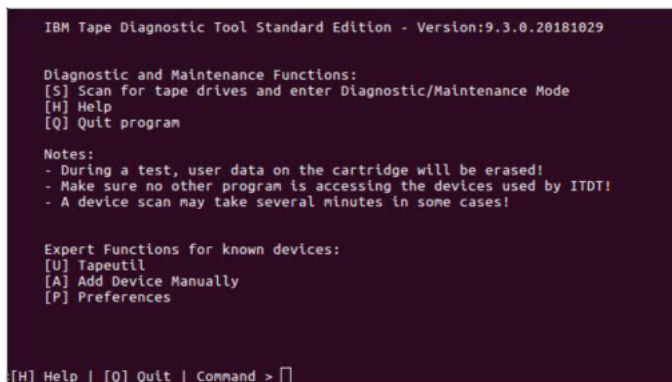
TS3310 tape libraries

Library Firmware Update with ITDT-SE and ITDT-GE is not supported by the TS3310 libraries. Update the firmware by using the Web User Interface for those libraries.

Standard Edition - Start menu commands

After program startup, ITDT-SE displays the start screen menu.

Figure 1. Start screen menu



```
IBM Tape Diagnostic Tool Standard Edition - Version:9.3.0.20181029

Diagnostic and Maintenance Functions:
[S] Scan for tape drives and enter Diagnostic/Maintenance Mode
[H] Help
[Q] Quit program

Notes:
- During a test, user data on the cartridge will be erased!
- Make sure no other program is accessing the devices used by ITDT!
- A device scan may take several minutes in some cases!

Expert Functions for known devices:
[U] Tapeutil
[A] Add Device Manually
[P] Preferences

[H] Help | [Q] Quit | Command > 
```

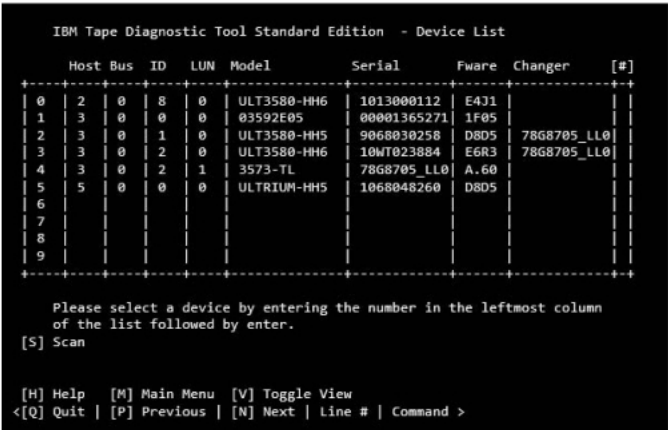
The following commands are available on the start screen menu:

- [S] Scan for tape drives and enter Diagnostic/Maintenance Mode
Opens the screen for the Scan function (refer to [Standard Edition - Scan menu commands](#)).
- [H] Help
Help starts and displays the available online help.
- [Q] Quit program
Quits the program.
- [U] Tapeutil
Opens the screen for the *Tapeutil* operation commands. These commands are the standardized *tapeutil* functions with most of the options available that were available with the previous tapeutil functions (refer to [Standard Edition - Tapeutil menu commands](#)).
- [A] Add Device Manually
Opens the screen for specifying a device manually instead of using the Scan function.
- [P] Preferences
Opens the dialog where default program settings can be defined and altered.

Standard Edition - Scan menu commands

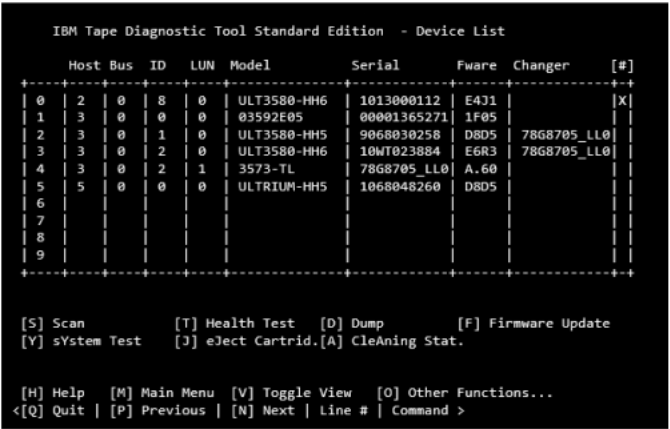
When ITDT-SE is used after **S** is entered on the start screen, the Scan function starts and displays the first device list screen.

Figure 1. Scan screen menu



To select a device, enter a number from the leftmost column, then click Enter.
Entering the **M** command returns to the Start Screen menu. Entering the **V** command toggles between displaying the physical device address and the driver name.

Figure 2. Scan screen second menu



This screen contains the **S**, **T**, **D**, **F**, **Y**, **J**, and **A** commands. Entering the **o** command displays the second device list screen. The second screen contains the **W**, **U**, **K**, **L**, **I**, **E**, **C**, and **R** commands.
The following commands are described:

- S - [Scan](#)
- T - [Health Test](#)
- D - [Dump](#)
- F - [Firmware Update](#)
- Y - [System Test](#)
- J - [Eject Cartridge](#)
- A - [Cleaning Statistics](#)

- O - [Other Functions](#)

Figure 3. More scan options

IBM Tape Diagnostic Tool Standard Edition - Device List

	Host	Bus	ID	LUN	Model	Serial	Fware	Changer	[#]
0	2	0	8	0	ULT3580-HH6	1013000112	E4J1		X
1	3	0	0	0	03592E05	00001365271	1F05		
2	3	0	1	0	ULT3580-HH5	9068030258	D8D5	78G8705_LL0	
3	3	0	2	0	ULT3580-HH6	10WT023884	E6R3	78G8705_LL0	
4	3	0	2	1	3573-TL	78G8705_LL0	A.60		
5	5	0	0	0	ULTRIUM-HH5	1068048260	D8D5		
6									
7									
8									
9									

[W] Full Write [U] Tape Usage [K] Check LTFS Readiness
[L] Library Test [I] Lib. Media Screening
[E] Encryption [C] Config.TCP/IP

[H] Help [M] Main Menu [V] Toggle View [R] Return
<[Q] Quit [P] Previous [N] Next Line # Command >

- W - [Full Write](#)
- U - [Tape Usage](#)
- K - [Check LTFS Readiness](#)
- L - [Library Test](#)
- I - [Library Media Screening](#)
- E - [Encryption](#)
- C - [Configure TCP/IP](#)
- R - [Return](#)

- [Scan](#)
- [Health Test](#)
- [Dump](#)
- [Firmware Update](#)
- [System Test](#)
- [Eject Cartridge](#)

The Eject Cartridge [J] function unloads a cartridge from a tape drive.

- [Cleaning Statistics](#)
- [Other Functions](#)
- [Full Write](#)
- [Tape Usage](#)
- [Check LTFS Readiness](#)
- [Library Test](#)
- [Library Media Screening](#)
- [Encryption](#)
- [Configure TCP/IP](#)
- [Return](#)

Scan

The Scan function [S] is used to discover all supported tape and library devices that are attached to the computer system so that they can be selected for the subsequent ITDT-SE operations. The scan function also serves as a connection test that can be used to verify correct attachment of the devices.

Make sure that no other program is accessing the devices that are used by ITDT-SE. For example, stop the backup jobs that are accessing the devices when ITDT-SE is used, or if not sure, stop the entire backup application.

After ITDT-SE is started, type **s** followed by Enter to activate the scan function.

Depending on the operating system and the number of attached drives, the scan can take several minutes. See [Standard Edition - known issues and limitations](#) for details.

During the scan operation, a bar in the lower left edge of the screen shows that the scan operation is still in progress.

When the scan is finished, the first device list screen is displayed.

Figure 1. Device List screen

Device number

IBM Tape Diagnostic Tool Standard Edition - Device List

	Host	Bus	ID	LUN	Model	Serial	Fware	Changer	[#]
0	2	0	8	0	ULT3580-HH6	1013000112	E4J1		X
1	3	0	0	0	03592E05	00001365271	1F05		
2	3	0	1	0	ULT3580-HH5	9068030258	D8D5	78G8705_LL0	
3	3	0	2	0	ULT3580-HH6	10WT023884	E6R3	78G8705_LL0	
4	3	0	2	1	3573-TL	78G8705_LL0	A.60		
5	5	0	0	0	ULTRIUM-HH5	1068048260	D8D5		
6									
7									
8									
9									

Name of current screen

If marked with an X, the device is selected

Commands

[S] Scan [T] Health Test [D] Dump [F] Firmware Update
[Y] System Test [J] Eject Cartrid. [A] Cleaning Stat.

[H] Help [M] Main Menu [V] Toggle View [O] Other Functions...
<[Q] Quit [P] Previous [N] Next Line # Command >

Command prompt

The first device list screen shows all detected devices and the connection information (host adapter number, bus number, SCSI/FCP ID, and LUN or driver name) along with product data (Model name, Unit Serial number, Microcode revision). For drives that are attached to a library, the Changer column shows the serial number of the changer the drive is attached to.

Scrollable data is indicated by "VVVVVVVV" in the bottom border of the table. To view the non-displayed entries, type + and press Enter.

Note: For fast down scrolling, type + followed by a space and the number of lines to scroll down, then press Enter. Alternately, type N and press Enter to scroll down one page.

To scroll back, use - instead of +.

Note: For fast up (backward) scrolling, type - followed by a space and the number of lines to scroll up. Press Enter, or type P and press Enter to scroll up one page.

If no devices appear or if devices are missing in the list, make sure that

- ITDT-SE is running with administrator/root rights.
- The devices are properly attached and powered on.
- Linux: The devices must be attached at boot time.
- i5/OS: Only tape drives are detected.
- Solaris, when no IBM® tape device driver is in use:
 - Ensure that sgen is correctly configured.

`file /kernel/drv/sgen.conf`

is correctly configured (see [Starting ITDT-SE on Solaris operating systems](#)).

- Solaris 10, see (see [Standard Edition - known issues and limitations](#))
- More than 10 devices displayed - scroll down the Device List.

ITDT-SE uses the IBM Tape Device Driver for its operations. If no IBM Tape Device Driver is installed, the generic device driver for the operating system is used instead. On Microsoft Windows, any Tape Device Driver that is installed is used.

If you must bypass the IBM Tape Device Driver for diagnostic purposes, start ITDT-SE with the following command.

`itdt -force-generic-dd`

Note: For operating system-specific information on how to use this command, see the corresponding Initial Startup sections.

When the wanted device is displayed, select the device for test. Only one device can be selected at a time.

Health Test

The Health Test function [T] checks if the tape device is defective and outputs a pass/fail result.

Attention: The health test function erases user data on the cartridge that is used for the test.

For the library or autoloader test, the Library Test [L] must be selected.

Note:

1. The test can take from 15 minutes up to 2 hours.
2. The test runs only on tape drives, not on autoloaders or libraries.

To complete the test function, it is recommended that a new or rarely used cartridge is used. Scaled (capacity-reduced) cartridges must not be used to test the device.

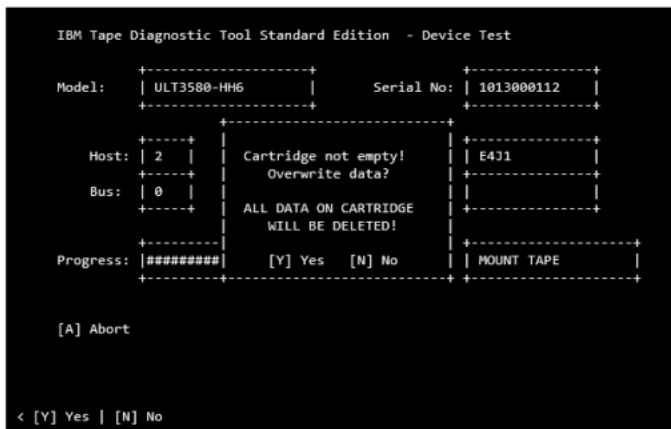
To test tape drives within a library, the library must be in online mode.

1. Start ITDT-SE, then type S and press Enter to scan for the devices.
2. Select the device that you want to test by entering its number and press Enter.
3. Type T followed by Enter to activate the test.

If no cartridge is inserted, ITDT-SE prompts to insert a cartridge. Either insert a cartridge and press Enter or stop the test by entering C followed by Enter.

Note: If ITDT-SE detects data on the cartridge, the Device Test screen displays a message (as shown in [Figure 1](#)).

Figure 1. Data Delete question

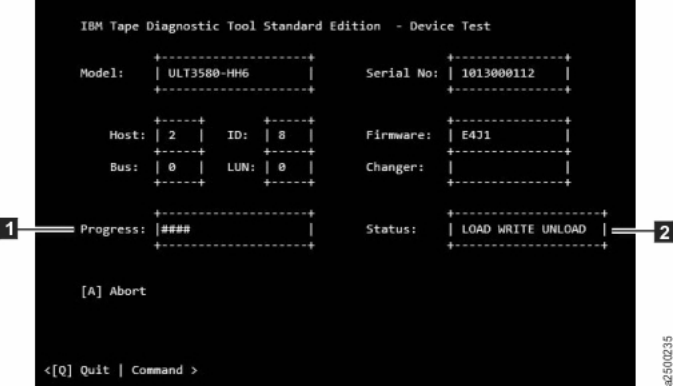


Type Y followed by Enter to continue the test if you are sure that data on the cartridge can be overwritten. If you are unsure, type N followed by Enter to stop the test.

During the test, the program shows a progress indicator in the form of a bar of number signs (#) (1) that shows the progress of a single subtest and also a description of that subtest. The user might stop the test by selecting the [A]Abort option (exception: POST A).

During the test, a progress indicator (1) is shown on the test screen. Messages from the test steps are shown in the Status field (2).

Figure 2. Test running

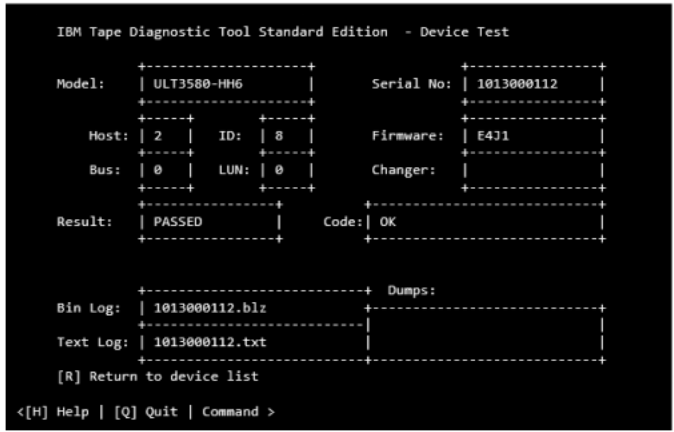


The test sequence contains the following steps.

1. Initialize Device
2. Read Thermal Sensor (might get skipped)
3. Mount Medium
4. [Medium Qualification] - only if the previous step indicated this requirement
5. Load/Write/Unload/Read/Verify
6. POST A
7. Performance Test (run 2 times if first run failed with performance failure)
8. Unmount Medium
9. Read Thermal Sensor (might get skipped)
10. Get FSC
11. Get Logs

The test can be stopped by typing `A` followed by Enter at any time except during the POST test, which is not interruptible.
Note: It might take some time until the test stops.

Figure 3. Test results



When all subtests are finished, ITDT-SE shows a screen that displays the attachment and device information as in the first device list screen. It also shows the test result and failure information in the code field. The screen also shows the output files that were generated during the test run. The files might be requested by the IBM® Support Center.

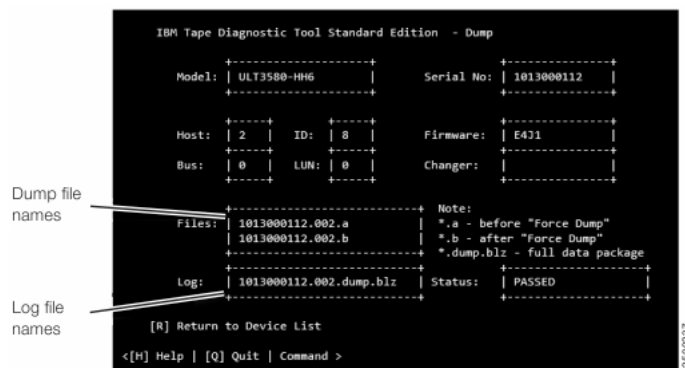
If you want to use other ITDT-SE functions, type `R` followed by Enter to return to the first device list screen. Otherwise, type `Q` followed by Enter to exit the program.

Dump

Complete the following steps to start the Dump [D] process.

1. Start ITDT-SE, then type `S` and press Enter to scan for the devices.
2. Select the device that you want to retrieve a dump from by entering its number and pressing Enter.
3. Type `D` and press Enter to start the dump retrieval for the selected device. The ongoing dump process is completed (it takes less than 1 minute).

Figure 1. Dump



When the dump process is completed on a tape library or autoloader other than the 3584/TS3500/TS4500, the Dump function stores 1 log file in the output folder of the program (*.blz). For the 3584/TS3500/TS4500, a dump file (*.a) is stored in the output folder.

Note: When the Dump function is completed for tape libraries or autoloaders other than the 3584/TS3500/TS4500, the log file contains only Log Sense and Mode Sense pages, while a Drive or 3584/TS3500/TS4500 dump contains much more diagnostic information.

Retrieve the files from the ITDT-SE output subdirectory that was created during the installation. The following are examples of the directory:

- Example output directory (Windows): c:\ITDT\output
- Example output directory (UNIX): /home/user/ITDT/output
- Example output directory (i5/OS): /home/ITDT/output
(On the IFS) use FTP or the System i Navigator to transfer the file

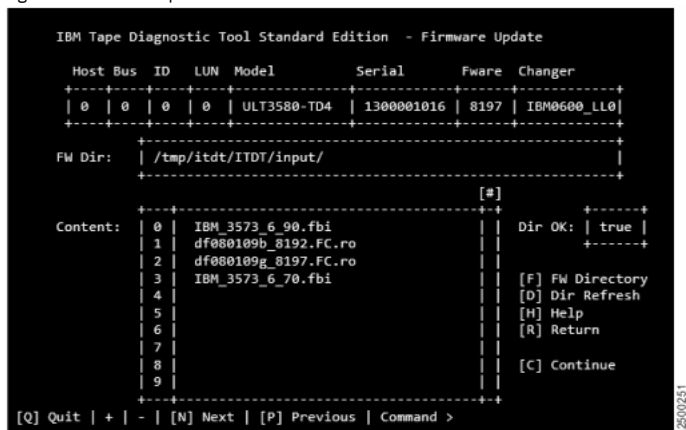
If you want to use other ITDT-SE functions, type **R** followed by Enter to return to the device list; otherwise, type **Q** followed by Enter to exit the program.

Firmware Update

The Firmware Update [F] upgrades the firmware of tape drives and tape libraries.

Note: See [con_a88u9_itdt_se_limits_all.html#con_a88u9_itdt_se_limits_all_3576](#) for information on how to update the firmware on TS4500 tape libraries.

Figure 1. Firmware Update screen



The following site is available for the latest firmware files: <http://www.ibm.com/systems/support/storage/tape>

Download the files to the ITDT-SE input subdirectory that was created during the installation. The following are examples of the directory:

Example input directory (Windows): c:\ITDT\input

Example input directory (Unix): /home/user/ITDT/input

Example input directory (i5/OS): /home/ITDT/input
(on the IFS) use FTP or the i-Series Navigator to transfer the file

To do a Firmware Update, complete the following steps:

1. Start ITDT-SE, then type **S** and press Enter to scan for the devices.
2. Select the device that you want to update by typing the number of the device and pressing Enter.
3. Type **F** and press Enter to display the Firmware Update screen.
4. To select the needed firmware update, complete one of the following steps:
 - If the downloaded firmware file is listed in the Content field of the Firmware Update screen, type the corresponding line number and press Enter.
 - If the firmware file is stored in a directory other than FW Dir, type **F** followed by a space and the fully qualified path to the directory that contains the firmware file, then press Enter.

For example, enter the following to change the firmware directory (UNIX):

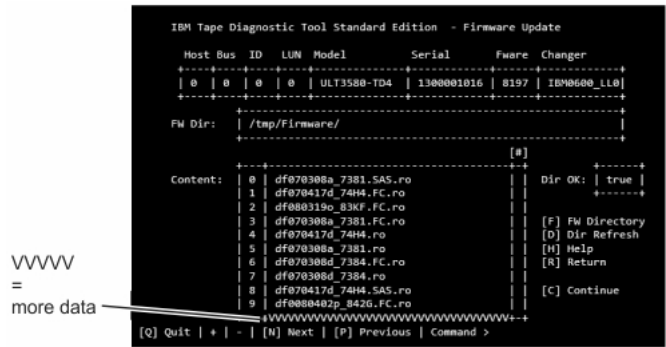
```
f /home/user/firmware
```

- If no files are displayed in the Content field, check the Dir OK field on the right side of the screen. It indicates true if the directory exists, false otherwise.
- If the content of the displayed FW Dir changed, type **D** and press Enter to refresh the directory content.

Note: The selected file name is reset to the first item (#0) after the Refresh function is used.

- If the displayed directory contains more files than the files shown, type + and press Enter to scroll down the list. For fast down scrolling type + followed by a space and the number of lines to scroll down then press Enter. To scroll back, use – instead of +. Scrollable data is indicated by "VVVVVVVVVVVVVVVVVVVV".

Figure 2. Scrollable Data screen



5. After the firmware file is selected, type **C** and press Enter to continue.
6. Before the firmware update is started, make sure the file that is displayed in the FW File field is the correct file.
 - If the correct file is displayed, proceed to the next step.
 - If the correct file is not displayed, type **C** and press Enter to change the selected firmware file. Go to Step 4.

Note: The selected file name is reset to the first item in the list when you return to that dialog from the Start Update dialog.

7. If you decide to run the firmware update, type **S** and press Enter to start the firmware update. During the firmware update, a firmware update progress screen is displayed.

Attention: Once started, do not interrupt the firmware update.

The firmware update usually takes 3-5 minutes, but it can take up to 45 minutes for libraries. If you decide not to run the firmware update, type **R** and press Enter to return to the Device List.

Note: If ITDT-SE detects FIPS-certified drive firmware, it displays a warning dialog. Before you continue, ensure that you use FIPS-certified firmware to update the drive.

8. After completion, the Status field on the lower right side indicates PASSED if the firmware was updated successfully and FAILED otherwise. Type **R** and press Enter to return to the Device List.

System Test

The System Test [Y] is a short test that completes the following steps:

- Reveals system performance bottlenecks. Compressible data throughput values can reveal bandwidth limitations that are caused by the system, cabling, or HBA.
- Measures performance variations across the different block sizes to find the ideal block size for the system configuration.

The System Test runs only on tape drives, not on autoloaders or libraries. To complete a System Test on tape drives within a library, the library must be in online mode.

1. Start ITDT-SE, type **S**, and press Enter to scan for the devices.
2. Type **Y** and press Enter to start the System Test. ITDT-SE then switches to the System Test screen. If no cartridge is inserted, ITDT-SE prompts to insert a cartridge. Either insert a cartridge and press Enter or stop the test by typing **C** followed by Enter.

Note: If ITDT-SE detects data on the cartridge, it shows the System Test screen, and displays the following message.

Cartridge not empty!
Overwrite data?

Type **Y** followed by Enter to continue the test if you are sure that data on the cartridge can be overwritten. If you are unsure, type **N** followed by Enter to stop the test.

The System Test is completed as follows:

- a. System Test determines the amount of data to write for each supported blocksize (a percentage of the cartridge is written for each blocksize)
- b. The test determines the maximum supported blocksize of the system – maximum is 8 MiB (generic 1 MiB).
- c. Next, the System Test writes the calculated size with supported block sizes in powers of two down to 64 kB (at maximum 8192, 4096, 2048, 1024, 512, 256, 128, 64), first with incompressible data, then with compressible data. Five different block sizes are written.
- d. At the end of the test, a summary screen is displayed.

Figure 1. System Test results

```

IBM Tape Diagnostic Tool Standard Edition - System Test

Model      Serial      Fware      Result:
+-----+-----+-----+-----+
| ULT3580-TD4 | F09789A000 | 89B2 | | PASSED |
+-----+-----+-----+-----+

Compress- Transfer Data Size Elapsed Data Rate
sible    Size (KB)  (MB)    Time (s)  (MB/s)
+-----+-----+-----+-----+
| No      | 512          | 953     | 12.684    | 75.1734   |
| No      | 256          | 953     | 13.245    | 71.9894   |
| No      | 128          | 953     | 13.26     | 71.9174   |
| No      | 64           | 953     | 13.256    | 71.9391   |
| Yes     | 512          | 953     | 4.501     | 211.842   |
| Yes     | 256          | 953     | 4.922     | 193.722   |
| Yes     | 128          | 953     | 6.279     | 151.875   |
| Yes     | 64           | 953     | 9.002     | 105.935   |
+-----+-----+-----+-----+

Code:
+-----+
| OK |
+-----+

Text Log:
+-----+
| 1013000112.004.txt |
+-----+

Bin Log:
+-----+
| 1013000112.004.blz |
+-----+

<[Q] Quit | + | - | [R] Return>

```

"Compressible = Yes" means that the data written was just zeros so that the data is compressed by the drive with a maximum compression ratio. "Compressible = No" means that a data pattern was written that the drive almost cannot compress at all. If the compression ratio is 1, the drive was not able to compress the data (equivalent to 1:1 compression ratio). If the compression ratio is 94.0, the drive was able to do 94:1 compression, meaning that 94 bytes in the original data is compressed to 1 byte on the medium. 100.0 means 100 bytes is compressed down to 1 byte on the medium.

The System Test can be stopped by typing **A** followed by Enter at any time.
 Note: It can take some time until the System Test stops.

If you want to use other ITDT-SE functions, type **R** followed by Enter to return to the device list. Otherwise, press **Q** followed by Enter to exit the program.

Eject Cartridge

The Eject Cartridge [J] function unloads a cartridge from a tape drive.

1. Start ITDT-SE, then type **S** and press Enter to scan for the devices.
2. Select the device that you want to unload a cartridge from by entering its number and pressing Enter.
3. Type **J** and press Enter to unload a cartridge.

Cleaning Statistics

Cleaning Statistics [A] retrieves statistical data about cleaning actions. Some devices do not support this function.

1. Start ITDT-SE, type **S**, and press Enter to scan for the devices.
2. Select the device that you want to retrieve cleaning statistics from by entering its number and pressing Enter.
3. Type **A** and press Enter to start the cleaning statistics retrieval for the selected device.

Other Functions

Other Functions [O] - type **O** followed by Enter to display a screen with the following commands:

- W - [Full Write](#)
- U - [Tape Usage](#)
- K - [Check LTFs Readiness](#)
- L - [Library Test](#)
- I - [Library Media Screening](#)
- E - [Encryption](#)
- C - [Configure TCP/IP](#)
- R - [Return](#)

Full Write

The Full Write [F] function writes the entire cartridge with a specified block size either with compressible or incompressible data and output performance data.

Attention: The Full Write function erases data on the cartridge that is used for the test.
 Note:

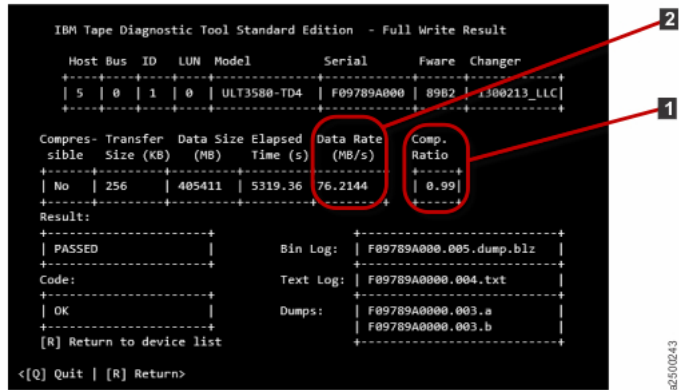
1. The Full Write function takes approximately 2 hours when incompressible data is written, less time for compressible data.
2. The Full Write function runs only on tape drives, not on autoloaders or libraries.

The Full Write test can be used to

- Demonstrate that the drive can write the full amount of data on a cartridge.

- Identify system issues with compression.
Drive data compression is always turned on during the full write. When run with compressible data, the output shows the compression rate. If the compression rate is higher than 1.0 but the system does not appear to be able to compress data on the cartridge, check the device driver and software settings to see whether they disable compression.
1. After ITDT-SE is started, type `s` followed by Enter to activate the device scan.
 2. Select the device that you want to write to by entering its number and press Enter.
 3. Type `w` and press Enter to start the full write.
ITDT-SE then switches to the Full Write screen. If no cartridge is inserted, ITDT-SE prompts to insert a cartridge. Either insert a cartridge and press Enter or stop the test by typing `c` followed by Enter.
- Note: If ITDT-SE detects data on the cartridge, it shows the Full Write screen, and displays the following message:
- ```
Cartridge not empty!
Overwrite data?
```
- Type `y` followed by Enter to continue the test if you are sure that data on the cartridge can be overwritten. If you are unsure, type `N` followed by Enter to stop the test.
4. The system prompts for entry of a transfer size between 32 KB and the maximum block size that is supported by the system (maximum value is 8 MiB). This action is a check for the type of supported block size that is completed. Enter the appropriate values for your system.  
Note: Values of 16 KB and 32 KB are not tested in cases where the capability of a system supports higher block sizes.
  5. Select the type of data to write, either [C] Compressible or [I] Incompressible.  
During the full write, the program shows a progress indicator in form of a bar of number signs (#) that shows the progress of the full write.
- The full write can be stopped by typing `A` followed by Enter at any time.
- Note: It can take some time until the full write stops.
- If all write operations are finished, ITDT-SE shows a screen that displays the compression ratio ( 1 ) and the write performance (shown in 2 as the Data Rate) for the selected block size. If an error occurred during the full write, data is only written partially.
- "Compressible = Yes" means that the data written was just zeros so that the data is compressed by the drive with a maximum compression ratio. "Compressible = No" means that a data pattern was written that the drive almost cannot compress at all. If the compression ratio is 1, the drive was not able to compress the data (equivalent to 1:1 compression ratio). If the compression ratio is 94.0, the drive was able to do 94:1 compression, meaning that 94 bytes in the original data is compressed to 1 byte on the medium. 100.0 means 100 bytes is compressed down to 1 byte on the medium.
- If you want to use other ITDT-SE functions, type `R` followed by Enter to return to option 4 the device list. Otherwise, type `Q` followed by Enter to exit the program.

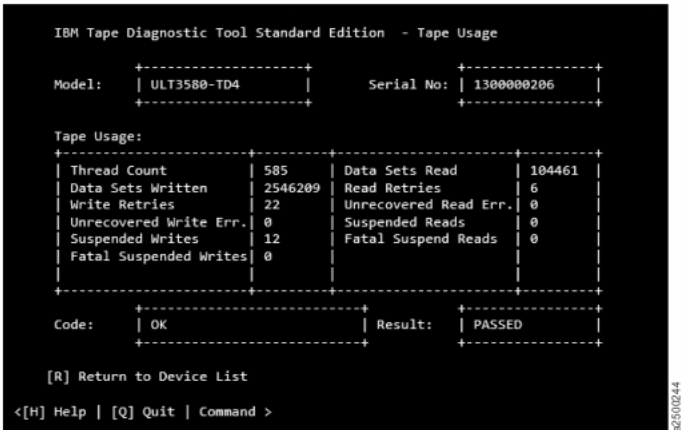
Figure 1. Full Write results



## Tape Usage

The Tape Usage [U] function retrieves statistical data and error counters from a cartridge.

Figure 1. Tape Usage screen





1. After ITDT-SE is started, type **S** followed by Enter to activate the device scan.
2. Select the device that you want to test by entering its number and press Enter.
3. Type **U** followed by Enter to start the tape usage log retrieval. ITDT-SE then switches to the tape usage screen. If no cartridge is inserted, ITDT-SE prompts to insert a cartridge. Either insert a cartridge and press Enter or stop the test by entering **C** followed by Enter.

During the get logs operation, the program shows a progress indicator in form of a bar of number signs (#) that shows the progress of a single suboperation and a description of that operation.

When all suboperations are finished, ITDT-SE shows a Tape Usage completion screen. The Status field on the lower right side indicates PASSED if the log retrieval completed successfully and ABORTED otherwise.

---

## Check LTFS Readiness

The Check LTFS Readiness test analyzes the operating system and tape drive environment to ensure that the IBM® linear tape file system can be installed. This test checks the operating system version, the tape device driver version, the tape drive firmware, and the LTFS HBA requirements.

Note: The tape drive firmware must be at least version: C7RC (for LTO 5), C974 (for LTO 6), and 36A5 (for TS1140 and TS1150). The LTFS Readiness Check requires an empty data cartridge. The maximum transfer size is detected at the beginning of the test. If it is under 512kB, a warning is displayed. If it is under 256kB, the test fails. The LTFS Readiness Check can return with result FAILED and one of the following error codes.

Table 1. Codes and root causes

| Code                    | Root causes                                      |
|-------------------------|--------------------------------------------------|
| LTFS CDB LENGTH         | Unsupported SCSI 16 Byte command.                |
| LTFS CARTRIDGE TYPE     | Cartridge Density Code is not supported.         |
| LTFS XFER SIZE          | Supported block size is too less.                |
| LTFS PARTITION SUPPORT  | Partitioning commands are not supported.         |
| LTFS FW LEVEL           | Unsupported Tape Device firmware level.          |
| LTFS DATA TRANSFER SIZE | Data Integrity Test failed.                      |
| LTFS DD VERSION         | IBM Tape Device Driver version is not supported. |
| LTFS OPERATING SYSTEM   | Operating system is not supported.               |

---

## Library Test

The Library Test [L] starts and monitors the library-internal self-test. This test runs only on libraries and autoloaders, not on tape drives.

1. Start ITDT-SE, type **S**, and press Enter to scan for the devices.
2. Type **O** and press Enter to display the second device list screen.
3. On the second device list screen, type **L** and press Enter to start the Library Test.  
A Device Test screen is displayed and a functionality test on the tape library is completed.

At the end of the test, a results screen is displayed.

The Library Test can be stopped by typing **A** followed by Enter at any time.

Note: It can take some time until the Library Test stops.

If you want to use other ITDT-SE functions, type **R** followed by Enter to return to the device list; otherwise press **Q** followed by Enter to exit the program.

---

## Library Media Screening

Library Media Screening [L] generates dumps for each drive and cartridge within a library. It runs only on libraries (except TS3500/TS4500) and auto-loaders, not on tape drives.

First, the test tries to read dump files from each drive that is installed from the library. After that, the customer must select one drive for loading the cartridges.

All cartridges of the I/O and storage slots are moved - one after the other - from their source to the selected drive. A dump is taken and moved back to the source address.

In the result screen, the dumps taken and the count of dumps are displayed.

---

## Encryption

The Encryption [E] function is used to verify whether data on the cartridge was written encrypted. It reads both decrypted and raw data from the cartridge into two separate files on disk. The user can then verify that the data differs to ensure that encryption worked.

The Encryption function does not provide a Write - Read test.

The Encryption function is supported only on encryption enabled drives. It requires that an encryption infrastructure, including the Encryption Key Manager (EKM), is properly set up. An encrypted data cartridge must be used.

The Encryption function is supported for the following encryption environments:

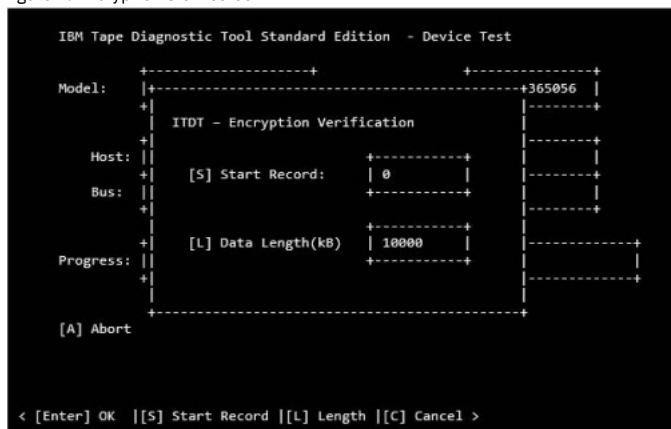
- System Managed: IBM® tape device driver must be installed and in use by ITDT to read decrypted data

- Library Managed
- Application Managed: Only raw encrypted data is read (result file \*.ENC)

Note: On i5/OS, media changers and media changer operations are not supported by this release of ITDT-SE. To test a tape drive inside a library, the tape drive must be varied online and the tape library must be varied offline (see [Starting ITDT-SE on i5/OS operating systems](#) for details). As the library is varied offline, the Encryption function does not deliver decrypted data in a Library Managed Encryption environment.

1. After ITDT-SE is started, type **S** followed by Enter to activate the device scan.
2. Select the device that you want to test by entering its number and press Enter.
3. Type **E** and press Enter to start the encryption test. ITDT-SE then switches to the Encryption Verification screen. On this screen, the system requires the entry of the number of the start record and the amount of data (in KB) to be read.
4. Type **S** followed by a space and the start record number, then press Enter to enter the start record number. Type **L** followed by a blank and the data length, then press Enter to enter the data length, maximum 100000 KB.

Figure 1. Encryption Start screen



5. If you entered the values correctly, press Enter to start the encryption. During the encryption, the program shows a progress indicator in form of a bar of number signs (#) that shows the progress of a single subtest and information about that subtest.

The Encryption function can be stopped by typing **A** followed by Enter at any time.

Note: It can take some time before the Encryption function stops.

If all encryption operations are finished, ITDT-SE shows a screen that displays the Status field on the lower left side that indicates PASSED if the encrypted test completed successfully and ABORTED otherwise.

The screen also shows the output files that were generated during the Encryption function:

- file *serial#*.n.ENC contains the raw encrypted data
- file *serial#*.n.DEC contains the decrypted data

[Table 1](#) defines the abort codes.

Table 1. Abort code definitions

| ABORT CODE                | ROOT CAUSE                                                                                                         |
|---------------------------|--------------------------------------------------------------------------------------------------------------------|
| LOCATE FAILED             | Start position as requested by the user was not reached                                                            |
| MEDIUM NOT ENCRYPTED      | ITDT detected medium as non-encrypted                                                                              |
| NO DRIVER SPECIAL FILE    | System-Managed environment, but generic device file is used instead of IBM device driver special file              |
| DRIVE ENCRYPTION DISABLED | Mode Sense detected disabled drive encryption                                                                      |
| UNEXPECTED DATA           | <ul style="list-style-type: none"> <li>• Set Raw read mode failed</li> <li>• One of the commands failed</li> </ul> |
| END OF MEDIUM             | End of medium that is encountered before the specified amount of data was read                                     |
| END OF DATA               | End of data that is encountered before the specified amount of data was read                                       |
| READ FAILED               |                                                                                                                    |
| ENCRYPTION ERROR          |                                                                                                                    |
| INVALID PARAMETER         | User entered data length of 0 kB                                                                                   |
| FILE IO ERROR             | The hard drive that ITDT is installed on might have run out of space.                                              |

If you want to use other ITDT-SE functions, type **R** followed by Enter to return to the device list. Otherwise, type **Q** followed by Enter to exit the program.

## Configure TCP/IP

Configure TCP/IP [C] configures the ethernet port settings of LTO 5, TS1140, and later drives. For those drives, the current settings are read and displayed and can be changed.

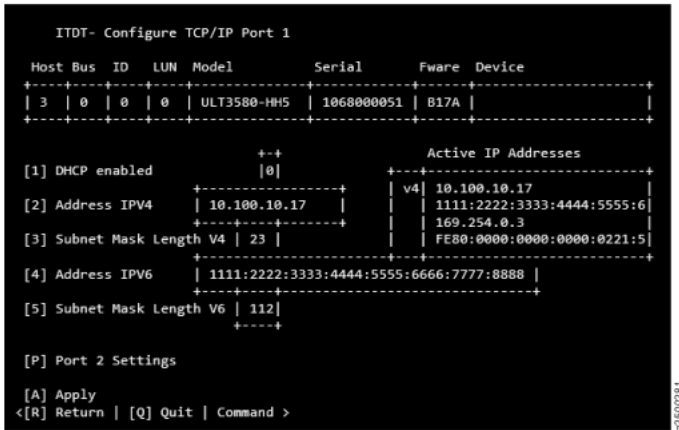
Note: LTO drives have one port and TS1140 and later drives have two ports that can be configured. Configuring the ethernet ports must not be done in a TS3500/TS4500. Although the ports can be configured, it is ineffective.

1. Start ITDT-SE, type **S**, and press Enter to scan for the devices.
2. Select a device from the list (just the ones that are listed are supported) by entering the number, then press Enter.
3. Type **O** and press Enter to display the second device list screen.

4. On the second device list screen, type **C** and press Enter to open the Configure TCP/IP screen.

ITDT-SE switches to the Configure TCP/IP screen and reads the data configuration of port 1. To toggle between port 1 and 2, type **P** and press Enter.

Figure 1. TCP/IP screen: Read data



Each parameter can be set by entering the number (1 - 5) and a following value. If you want to enable DHCP, enter '1 1' and press Enter. The value of the DHCP field is refreshed with the value entered.

The values for both ports are applied to the drive by entering **A** and pressing Enter. ITDT-SE configures the drive and the current active addresses are shown in the field: Active IP Addresses. Regular field values:

```
[1] DHCP enabled: 0/1 (false/true)
[2] Address IPV4 Regular IPv4 address
[3] Subnet Mask Length V4: 0...23
[4] Address IPV6: Regular IPv6 address
[5] Subnet Mask Length V4: 0...127
```

If you want to use other ITDT-SE functions, type **R**, then press Enter to return to the device list. Or, press **Q** and Enter to exit the program.

## Return

Return **[R]** - type **R** followed by Enter to go back to the first device list screen.

## Standard Edition - Tapeutil menu commands

When the user runs the **U** command on the ITDT-SE start screen, the Tapeutil operation screen is displayed.

Note: On any screen, to start a command, press the shortcut displayed in brackets [ ], followed by Enter.

The following commands are described in this section:

- [\[1\] Open a Device](#)
- [\[2\] Close a Device](#)
- [\[3\] Inquiry](#)
- [\[4\] Test Unit Ready](#)
- [\[5\] Reserve Device](#)
- [\[6\] Release Device](#)
- [\[7\] Request Sense](#)
- [\[8\] Log Sense](#)
- [\[9\] Mode Sense](#)
- [\[10\] Query Driver Ver. \(Version\)](#)
- [\[11\] Display All Paths](#)
- [\[12\] Query Runtime Info](#)
- [\[20\] Rewind](#)
- [\[21\] Forward Space File Marks](#)
- [\[22\] Backward Space File Marks](#)
- [\[23\] Forward Space Records](#)
- [\[24\] Backward Space Records](#)
- [\[25\] Space to End of Data](#)
- [\[26\] Read and Write Tests](#)
- [\[27\] Read or Write Files](#)
- [\[28\] Erase](#)
- [\[29\] Load Tape](#)
- [\[30\] Unload Tape](#)
- [\[31\] Write File Marks](#)
- [\[32\] Synchronize Buffers](#)
- [\[33\] Query/Set Parameter](#)
- [\[34\] Query/Set Tape Position](#)
- [\[35\] Query Encryption Status](#)
- [\[36\] Display Message](#)

- [\[37\] Report Density Supp.\(Support\)](#)
- [\[38\] Test Encryp. Path \(Test Encryption Key Path/Setup\)](#)
- [\[39\] Configure TCP/IP Port](#)
- [\[50\] Element Information](#)
- [\[51\] Position to Element](#)
- [\[52\] Element Inventory](#)
- [\[53\] Exchange Medium](#)
- [\[54\] Move Medium](#)
- [\[55\] Initialize Element Status](#)
- [\[56\] Prevent/Allow Medium Removal](#)
- [\[57\] Initialize Element Status Range](#)
- [\[58\] Read Device IDs](#)
- [\[59\] Read Cartridge Location](#)
- [\[70\] Dump/Force Dump/Dump](#)
- [\[71\] Firmware Update](#)
- [\[101\] HDP Discover](#)
- [\[102\] HDP Initiate Call Home](#)
- [\[103\] HDP Show Import Export Elements](#)

## [1] Open a Device

When you select the **Open a Device** command [1]:

1. ITDT checks if a device is already opened.
2. You are prompted for a device special file name.
3. You are prompted for an open mode (rw, ro, wo, append).
4. ITDT opens the device that you selected.

Note: Always use the Read Only mode when you are working with write-protected media.  
The combination of open cmd with parameter -force-generic-dd is not supported.

## Device file names - device addressing

ITDT supports generic and Device Driver claimed devices. This section shows examples for device names (addressing) of all supported platforms. The used abbreviations stand for:

|        |                                                                  |
|--------|------------------------------------------------------------------|
| host   | Number of the host adapter (SCSI, FC, SAS)                       |
| bus    | Number of the bus from the host adapter                          |
| target | Target Number of the device                                      |
| lun    | Logical Unit Number of the device that is separated with blanks. |

Note: The correct IDs are reported in the ITDT Control Center after a scan or with the scripting function "scan".

Table 1. Device addressing

|                   | IBM Tape Device Driver            | Generic IDs separated with blanks | Generic (alternative, as a result from ".itdt scan") |
|-------------------|-----------------------------------|-----------------------------------|------------------------------------------------------|
| IBM AIX           | /dev/rmtX.Y<br>/dev/smcX          | <host><bus><target><lun>          | H<host>-B<bus>-T<target>-L<lun>                      |
| Linux             | /dev/IBMtapeX<br>/dev/IBMchangerX |                                   |                                                      |
| Microsoft Windows | \\.\tape0<br>\\.\changer0         |                                   |                                                      |
| Oracle Solaris    | /dev/rmtXsmc<br>/dev/smc/Xchng    |                                   |                                                      |
| HP-UX             | /dev/rmt/Xmnb<br>/dev/rmt/Xchng   |                                   |                                                      |
| Apple Mac         | -                                 |                                   |                                                      |
|                   |                                   | <host><bus><target><lun>          | H<host>-B<bus>-T<target>-L<lun><br>tapeX<br>changerX |
| IBM 'i'           | -                                 | -                                 | Device Name; for example, TAP01                      |

Example for Linux (IBM Tape Driver devices):

```
./itdt scan
Scanning SCSI Bus ...
#0 /dev/IBMchanger2 - [3573-TL]-[C.50] S/N:00L4U78D6118_LL0 H3-B0-T5-L1 (IBM-Device)
#1 /dev/IBMtape3 - [ULT3580-TD5]-[BBNE] S/N:1168001104 H3-B0-T5-L0
Changer: 00L4U78D6118_LL0 (IBM-Device)
Exit with code: 0
```

Example for Linux (generic Operating System devices):

```
./itdt -force-generic-dd scan
Scanning SCSI Bus ...
#0 /dev/sg7 - [3573-TL]-[C.50] S/N:00L4U78D6118_LL0 H3-B0-T2-L1 (Generic-Device)
#1 /dev/sg8 - [ULT3580-TD5]-[BBNE] S/N:1168001104 H3-B0-T5-L0
Changer: 00L4U78D6118_LL0 (Generic-Device)
Exit with code: 0
```

To open the tape device by using the IBM Tape Device Driver, use the `/dev/IBMtape3` as device file name. By using H3-B0-T5-L0 or 3 0 5 0 as the device name, it is opened with the generic Operating System device driver.

---

## [2] Close a Device

When you select the **Close a Device** command [2]:

1. ITDT checks if the device is already closed.
2. ITDT closes the device.

---

## [3] Inquiry

When you select the **Inquiry** command [3]:

1. You are prompted for page code.
2. ITDT then displays a decoded format of a hexadecimal dump and prints a hexadecimal dump of the inquiry data.

---

## [4] Test Unit Ready

When you select the **Test Unit Ready** (TUR) command [4], ITDT issues the Test Unit Ready *ioctl* command.

---

## [5] Reserve Device

When you select the **Reserve Device** command [5], ITDT issues a reserve command for the device.

---

## [6] Release Device

When you select the **Release Device** command [6], ITDT issues a release command for the device.

---

## [7] Request Sense

When you select the **Request Sense** command [7]:

1. ITDT issues a **Request Sense** command.
2. ITDT then displays a decoded format of hexadecimal dump sense data and prints hexadecimal dump sense data.

---

## [8] Log Sense

When you select the **Log Sense** command [8]:

1. You are prompted for Log Sense Page.
2. ITDT issues a mode sense command.
3. ITDT completes a hexadecimal dump page.

---

## [9] Mode Sense

When you select the **Mode Sense** command [9]:

1. You are prompted for Mode Sense Page.
2. ITDT issues mode sense command.
3. ITDT completes a hexadecimal dump page.

---

## [10] Query Driver Ver. (Version)

When you select the **Query Driver Version** command [10]:

1. ITDT issues the required command to get the driver version.

2. ITDT prints the driver version.

---

## [11] Display All Paths

When you select the **Display All Paths** command [11]:

1. ITDT issues an *ioctl* command.
2. ITDT outputs decoded path information for all paths.

---

## [12] Query Runtime Info

When you select the **Query Runtime Info** command [12]:

1. ITDT issues the required command to get the runtime info.
2. ITDT prints out the Dynamic Runtime Attribute Values.

---

## [20] Rewind

When you select the **Rewind** command [20], ITDT issues the *ioctl* rewind command for the device.

---

## [21] Forward Space File Marks

When you select the **Forward Space File Marks** command [21]:

1. You are prompted for the number of file marks to space forward.
2. ITDT issues (extrinsic) *ioctl* command.

---

## [22] Backward Space File Marks

When you select the **Backward Space File Marks** command [22]:

1. You are prompted for the number of file marks.
2. ITDT issues (extrinsic) *ioctl* command.

---

## [23] Forward Space Records

When you select the **Forward Space Records** command [23]:

1. You are prompted for the number of records to space forward.
2. ITDT issues (extrinsic) *ioctl* command.

---

## [24] Backward Space Records

When you select the **Backward Space Records** command [24]:

1. You are prompted for the number of records to space backward.
2. ITDT issues (extrinsic) *ioctl* command.

---

## [25] Space to End of Data

When you select the **Space to End of Data** (EOD) command [25], ITDT issues the (extrinsic) *ioctl* command.

---

## [26] Read and Write Tests

When you select the **Read and Write Tests** command [26]:

- You are prompted for block size (If you press Enter, the default block size is 10240 bytes).

Note: If the block size is zero, variable mode is used. With a fixed block size, a data amount of (block size \* blocks) is transferred with a single operation. This process can get rejected if the total amount exceeds the transfer size the system can handle.

- You are prompted for the number of blocks per read/write (If you press Enter, the default number of blocks is 20).
- You are prompted for the number of repetitions (If you press Enter, the default number of repetitions is 1).

You can then select one of the following options:

- Read data from tape (to run Read only test)
- Write data to tape (to run Write-only test)
- Write/Read/Verify (to run Read and Write test)

ITDT runs the selected test. Then, it displays the transfer size and block size that is used for this test, the number of records read/written, and the total bytes transferred.

---

## [27] Read or Write Files

When you select the **Read or Write Files** command [27]:

- You are prompted to specify the file name of the source file (for Write test) or the destination file (for Read test).
- You are prompted for the number of records to be read.

You can then select one of the following options:

- Read File from Tape: ITDT reads a file from tape and stores data into a file with the specified file name.
- Write File to Tape: ITDT reads data from file with the specified file name and writes data to tape.

ITDT displays the number of records read/written, the transfer size, and the total bytes transferred.

---

## [28] Erase

When you select the **Erase** command [28], ITDT issues the (extrinsic) *ioctl* command to erase the cartridge.

For more information, refer to the Tapeutil Scripting Command at [erase](#).

---

## [29] Load Tape

When you select the **Load Tape** command [29], ITDT issues the load tape command.

---

## [30] Unload Tape

When you select the **Unload Tape** command [30], ITDT issues the unload tape command.

---

## [31] Write File Marks

When you select the **Write File Marks** command [31]:

1. You are prompted for the number of file marks to write.
2. ITDT issues the (extrinsic) *ioctl* command.

---

## [32] Synchronize Buffers

When you select the **Synchronize Buffers** command [32], ITDT issues the *ioctl* command.

---

## [33] Query/Set Parameter

When you select the **Query/Set Parameter** command [33]:

1. ITDT displays non-changeable parameters.  
Note: The list of non-changeable parameters is operating system specific.
2. ITDT displays changeable parameters.  
Note: The list of changeable parameters is operating system specific. For a list of changeable parameters, refer to [Table 1](#).
3. You are prompted for parameter to change.
4. ITDT requests prompt for parameter value (if required).
5. ITDT requests safety prompt (if required).

6. ITDT issues the *ioctl* command.

---

## [34] Query/Set Tape Position

When you select the **Query/Set Tape Position** command [34]:

1. ITDT displays the current position
2. You are prompted for a new position to set.
3. ITDT issues the **Set Position** *ioctl* command.

---

## [35] Query Encryption Status

When you select the **Query Encryption Status** command [35]:

1. ITDT issues **Get Encryption State** *ioctl* command.
2. ITDT displays encryption settings (Encryption capability, Encryption method, Encryption state).

---

## [36] Display Message

When you select the **Display Message** command [36]:

1. You are prompted for the text of message 0 (8 characters or less).
2. You are prompted for the text of message 1 (8 characters or less).
3. You are prompted for message type (msg0, msg1, flash0, flash1, alt).
4. ITDT issues the **Display Message** *ioctl* command.

Not all drives have a display. The 3592 drive is the only one that has display message capability. It is the only one with a display that is more than one character long. Eight is the limit of the characters on a display screen.

---

## [37] Report Density Supp (Support)

When you select the **Report Density Support** command [37]:

1. ITDT prints report status text for all supported media.
2. ITDT issues **Report Density Support** *ioctl* command to retrieve all supported media.
3. ITDT prints all requested reports. Data is printed in a decoded way. Scroll the screen to print each one of the following information:
  - Density name
  - Assigning organization
  - Description
  - Primary density code
  - Secondary density code
  - Write OK
  - Duplicate
  - Default
  - Bits per MM
  - Media Width
  - Tracks
  - Capacity (megabytes).
4. ITDT prints report status text for current media
5. ITDT issues **Report Density Support** *ioctl* command to retrieve current media
6. ITDT prints report data in a decoded way.

---

## [38] Test Encryp. Path (Test Encryption Key Path/Setup)

When you select the **Test Encryption Key Path/Setup** command [38]:

Note: Not supported for the HP-UX operating system.

1. ITDT prints status message that server configuration and connections are tested
2. ITDT issues the **Encryption Diagnostics** *ioctl* command, **Ping Diag**
3. ITDT prints number of servers available or error message
4. ITDT issues the **Encryption Diagnostics** *ioctl* command, **Basic Encryption Diag**
5. ITDT prints completion code or error message
6. ITDT issues the **Encryption Diagnostics** *ioctl* command, **Full Encryption Diag**
7. ITDT prints completion code or error message.

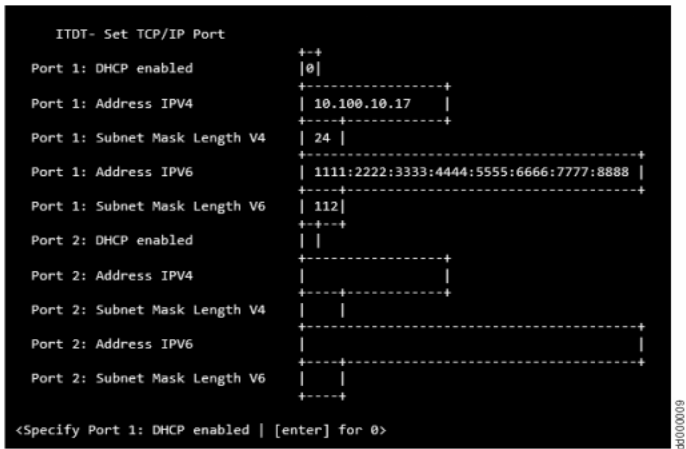
---

## [39] Configure TCP/IP Port



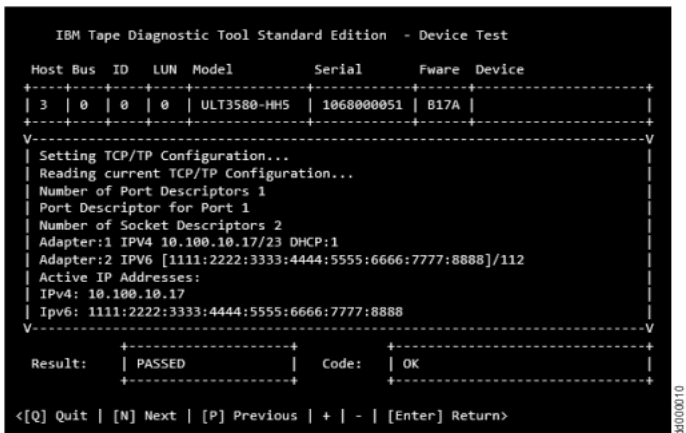
For LTO 5, TS1140, and later drives, the ethernet port settings can be configured with the **Configure TCP/IP Port** command. The **Configure TCP/IP Port** command displays the current settings:

Figure 1. TCP/IP Port command



You can change any of the settings by stepping through each parameter - press Enter. To change a parameter, enter the new value, then press Enter. All data is sent to the drive and the following screen display is shown:

Figure 2. TCP/IP Port command results



Note: Because earlier drive generations do not have an ethernet port, the **Configure TCP/IP Port** command is rejected for these devices with the following message:  
TCP/IP configuration is not supported on this product.

## [50] Element Information

When you select the **Element Information** command [50]:

1. ITDT issues the *iocctl* command
2. ITDT displays the
  - Number of robots
  - First robot address
  - Number of slots
  - First slot address
  - Number of I/E elements
  - First element address
  - Number of drives
  - First drive address

## [51] Position to Element

When you select the **Position to Element** command [51]:

1. You are prompted for destination address
2. ITDT issues the *iocctl* command.

## [52] Element Inventory

When you select the **Element Inventory** command [52]:

1. ITDT issues the **Element Info** *ioctl* command.
2. ITDT issues the **Element Inventory** *ioctl* command.
3. ITDT displays decoded element inventory information. Type *n* followed by Return to show the next page of information.

---

## [53] Exchange Medium

When you select the **Exchange Medium** command [53]:

1. You are prompted for source address.
2. You are prompted for first destination address.
3. You are prompted for second destination address.
4. ITDT issues the *ioctl* command.

---

## [54] Move Medium

When you select the **Move Medium** command [54]:

1. You are prompted for source address.
2. You are prompted for destination address.
3. ITDT issues the *ioctl* command.

---

## [55] Initialize Element Status

When you select the **Initialize Element Status** command [55]:

1. ITDT prints the command summary.
2. ITDT issues the *ioctl* command.

---

## [56] Prevent/Allow Medium Removal

When you select the **Prevent/Allow Medium Removal** command [56]:

1. You are prompted to select (1) for Prevent Removal, or (0) for Allow Removal.

---

## [57] Initialize Element Status Range

When you select the **Initialize Element Status Range** command [57]:

1. You are prompted for the first slot address.
2. You are prompted for the number of slots.
3. ITDT issues the *ioctl* command.

---

## [58] Read Device IDs

When you select the **Read Device IDs** command [58], ITDT retrieves the device ID information for all available drives and displays the information. Type *n* followed by Enter to show the next page.

---

## [59] Read Cartridge Location

When you select the **Read Cartridge Location** command [59]:

1. You are prompted for the address of the first slot.
2. You are prompted for number of elements.
3. ITDT verifies that the specified address range is valid, otherwise prints error message and exit.
4. ITDT issues the **Read Cartridge Location** *ioctl* command.
5. ITDT issues the **Element Info** *ioctl* command.
6. ITDT verifies that the address range is valid; otherwise, print the error message and exit.
7. If no slots are found in Element Info data, print the error message and exit.
8. ITDT issues the **Read Cartridge Location** *ioctl* command.
9. ITDT prints decoded storage element information, Type *n* followed by Enter to show next page.

---

## [70] Dump/Force Dump/Dump

When you select the **Dump/Force Dump/Dump** command [70]:

- ITDT retrieves the dump.
- ITDT issues the **Force Dump** command.
- ITDT retrieves second dump.
- ITDT displays the name of stored dump files and the output directory where they are stored. The dump file names start with the serial number of the device.

---

## [71] Firmware Update

When you select the **Firmware Update** command [71]:

1. ITDT displays the default input directory where the firmware files must be stored. The following are examples of the directory:
  - Example input directory (Windows): c:\itdt\input
  - Example input directory (UNIX): /home/user/itdt/input
2. You are prompted to specify a different input directory if required, or to press Enter to keep the standard ITDT directory.
3. You are prompted to specify the firmware file name and press Enter to start.
4. ITDT runs firmware update and displays progress status and result.  
Note: See [con\\_a88u9\\_itdt\\_se\\_limits\\_all.html#con\\_a88u9\\_itdt\\_se\\_limits\\_all\\_3576](#) for information on how to update the firmware on a TS4500 tape library.  
The following site is available for the latest firmware files: <http://www.ibm.com/storage/ito>.

---

## [101] HDP Discover

Discover the configuration of the HD-P system of all attached libraries - just TS3500/TS4500. The function scans the host for attached TS3500/TS4500 devices and uses their data to discover the HD-P system.

If it is successful, it delivers two maps that represents the logical and physical HD-P environment.

The logical map is a representation of the libraries, where a “1” is the indicator for a connection and a “-1” of no connection.

```
Command Result
+-----+
| Shuttle Call System discover.....
| Scanning devices.....
| DeviceFileName:/dev/smc1 SN:0000013AAA160404
|
| Discovering Libraries.....
| Passed
|
| Connection Map:
| LL01: /dev/smc1
|
| LL01
| LL01 -1
|
| Physical Map:
|
| SCS Discover passed
|
+-----+
< [Q] Quit | [N] Next | [P] Previous | + | - | [Enter] Return >
```

Supported platforms: AIX®, Linux® on P, and X64

Note: Mixed Media environments (3592 and LTO) are not supported.

---

## [102] HDP Initiate Call Home

Issuing a Test Call Home function.

Supported platforms: AIX®, Linux® on P, and X64

---

## [103] HDP Show Import Export Elements

Shows the extended HD-P Import Export Elements.

Supported platforms: AIX®, Linux® on P, and X64

---

## Add device manually

On the Start screen, press the [A] key followed by Enter to add a drive and a changer manually. In a dialog box, the tape device and the associated changer device can be defined. Supported formats are: devicename and H B T L. For a standalone tape drive the changer device is not required.

A default tape device is set and stepping through the device names the values can be altered.

When Enter is pressed, those drives which are accessible are opened. The manually added devices are shown in the same way as they are shown after an ITDT **Scan** command.

---

## Standard Edition - Program options

For problem determination and customization, ITDT is providing the following command line options.

```
-h -help Prints help information
-version Displays the version of ITDT,
 used configuration files and creation dates.
-force-generic-dd the usage of the generic Operating System driver
 (not using the IBM Tape Device driver) will be forced.
-LP logpath Use 'logpath' as logging path (default: output)
-L logfile Use 'logfile' for log messages (default: metro.log)

-LL Errors|Warnings|Information|Debug
 Set log level (default: Error)
-R resultdir Use 'resultdir' as result file path (default: output)
-settings Change default values of ITDT configuration parameters.
-start deviceaddress
-end deviceaddress Set range for device scan function, devicename and H B T L is supported.
```

---

## Standard Edition - Tapeutil scripting commands

Scripting is enabled with the 4.0 release of ITDT SE. ITDT-SE provides compatibility with earlier versions for existing tapeutil scripts. While some legacy commands exist, they are not documented in their entirety as they are phased out over time. New scripts must always use the scripts that are listed in this guide, the Common Command set (CS). Also, existing scripts must be modified for forward compatibility with ITDT.

You can find a list of commands, on each command you find the command, a description, parameter list, and which platforms are supported. Some commands have numbers after them. The numbers mean that a corresponding menu command is in [Standard Edition - Tapeutil menu commands](#).

The calling convention for the Common Command set is

```
itdt -f filename [Open Mode] Subcommand [Subcommand ...]
```

Note: "filename" is a device special file for the drive/changer or the device address (host bus target lun). For a complete list of the file name or address syntax, refer the section "Special Files" on each platform or go to [Device file names - device addressing](#).

The Open Mode flag is supported on all platforms. If the flag is not set, the device is opened in read/write mode. More parameters that might be required for opening the device are automatically detected and set.

```
-w mode
Open mode, by default Read/Write.
Valid modes are:
1 = Read/Write
2 = Read Only
3 = Write Only
4 = Append
```

Note: Always use the Read Only mode when you are working with write-protected media.

The new command set enables legacy commands on every platform, even if that is not previously supported by Tapeutil. The output follows current Tapeutil conventions. But, if different output displays for a single command on various platforms, the output is implemented according to the AIX® output as the primary scripting platform.

Tapeutil allows undocumented abbreviations for some of the commands. For example, it was possible to shorten "inquiry" to "inq" or "inqu" or "inqui". The following command abbreviations are supported by ITDT-SE too: inq(uiiry), req(sense), pos(ition), ele(mentinfo), inv(entory), devid(s), cartridge(location). Deprecated commands are listed at [Deprecated commands](#). Also, there is a list of unsupported commands and known exceptions at [Standard Edition scripting commands: known limitations and deviations](#).

The following commands are described.

General commands

- [allow](#)
- [devinfo](#)
- [inquiry|inq|inqj](#)
- [logpage|logpagej](#)

- [loop](#)
- [modepage](#)
- [prevent](#)
- [print](#)
- [grypath](#)
- [gryversion](#)
- [release](#)
- [regsense](#)
- [reserve](#)
- [scan/scanj](#)
- [sleep](#)
- [tur](#)
- [vpd](#)

#### Tape commands

- [append](#)
- [bsf](#)
- [bsr](#)
- [channelcalibration](#)
- [chgpart](#)
- [density](#)
- [display](#)
- [erase](#)
- [formattape](#)
- [fdp](#)
- [fdpl](#)
- [fsf](#)
- [fsr](#)
- [getparms](#)
- [idp](#)
- [idpl](#)
- [list](#)
- [load](#)
- [logsense](#)
- [grypar/grypart](#)
- [grylbp](#)
- [grypos](#)
- [grytcpip](#)
- [read](#)
- [resetdrive](#)
- [rmp](#)
- [runtimeinfo/gryruntimeinfo](#)
- [rewind](#)
- [rtest](#)
- [rwtest](#)
- [sdp](#)
- [sdpl](#)
- [seod](#)
- [setparm](#)
- [setpos](#)
- [settcpip](#)
- [sync](#)
- [unload](#)
- [verlbp](#)
- [weof](#)
- [write](#)
- [wtest](#)

#### Medium Changer Subcommands

- [audit](#)
- [cartridgeolocation](#)
- [elementinfo](#)
- [exchange](#)
- [inventory](#)
- [move](#)
- [position](#)

#### Service Aid commands

- [dump](#)
- [ekmtest](#)
- [encryption](#)
- [ucode](#)
- [tapephcp](#)
- [ltsphcp](#)
- [verify](#)
- [devicestatistics](#)
- [checkltsreadiness](#)
- [ltsdefragmentation](#)

- [standardtest](#)
- [fullwrite](#)
- [systemtest](#)
- [tapeusage](#)

#### HD-P commands

- [hdp\\_discover](#)
- [hdp\\_sendererror](#)
- [hdp\\_show](#)

#### Library RESTful commands

- [TS4500 REST over SCSI](#)
- [TS4300: RESTful API](#)
- [allow](#)
- [devinfo](#)
- [inquiry|inq|inqj](#)
- [logpage|logpagej](#)
- [loop](#)
- [modepage](#)
- [prevent](#)
- [print](#)
- [qrypath](#)
- [qryversion](#)
- [release](#)
- [reqsense](#)
- [reserve](#)
- [scan|scanj](#)
- [sleep](#)
- [tur](#)
- [vpd](#)
- [append](#)
- [bsf](#)
- [bsr](#)
- [channelcalibration](#)
- [chgpart](#)
- [density](#)
- [display](#)
- [erase](#)
- [formattape](#)
- [fdp](#)
- [fdpl](#)
- [fsf](#)
- [fsr](#)
- [getparms](#)
- [idp](#)
- [idpl](#)
- [list](#)
- [load](#)
- [logsense](#)
- [qrypar|qrypart](#)
- [qrylbp](#)
- [qrypos](#)
- [qrytcpip](#)
- [qrytemp](#)
- [read](#)
- [readattr](#)
- [resetdrive](#)
- [rmp](#)
- [runtimeinfo|qryruntimeinfo](#)
- [rewind](#)
- [rtest](#)
- [rwtest](#)
- [sdp](#)
- [sdpl](#)
- [seod](#)
- [setparm](#)
- [setpos](#)
- [settcpip](#)
- [sync](#)
- [unload](#)
- [verlbp](#)
- [weof](#)
- [write](#)
- [writeattr](#)
- [wtest](#)
- [audit](#)
- [cartridgelocation](#)
- [elementinfo](#)

- [exchange](#)
- [inventory](#)
- [move](#)
- [position](#)
- [dump](#)
- [ekmtest](#)
- [encryption](#)
- [ucode](#)
- [tapephcp](#)
- [tftsphcp](#)
- [verify](#)
- [devicestatistics](#)
- [checktftsreadiness](#)
- [tftsdefragmentation](#)
- [standardtest](#)
- [fullwrite](#)
- [systemtest](#)
- [tapeusage](#)
- [hdp discover](#)
- [hdp\\_senderror](#)
- [hdp show](#)
- [TS4500 REST over SCSI](#)
- [TS4300: RESTful API](#)
- [Deprecated commands](#)
- [Alternative mount/demount script - sample for Windows](#)
- [Standard Edition scripting commands: known limitations and deviations](#)

---

## allow

(Deprecated: **unlock, -o rem**) Allow medium removal for tape or changer devices (unlock door). The counter command for this is prevent.

Parameters:

**None**

Supported platforms: All

---

## devinfo

(Deprecated: **-o gdi**) Show device information (device type, sub type and block size)

Parameters:

**None**

Supported platforms: AIX, Solaris, HP-UX

---

## inquiry|inq|inqj

`inquiry` or `inq` command identifies the device and returns the output in hexadecimal format. `inqj` command is similar to `inquiry` or `inq` command but generates the output in JSON format.

Parameters:

**[Page code in Hex, 00-FF without leading x]**

Following is an example of an output from `inquiry` or `inq` command:

```
[root@franc ITDT]# ./itdt -f 4 0 8 0 inq
Issuing std inquiry...
Inquiry Data, Length 96
 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000 - 0180 0602 3301 1002 4942 4D20 2020 2020 [...]3...IBM
0010 - 3033 3539 3245 3037 2020 2020 2020 2020 [03592E07
0020 - 3343 3834 4A50 3030 3030 3133 4430 3031 [3C84JP000013D001]
0030 - 3233 2031 8100 1181 0113 0008 0000 0000 [23 1.....]
0040 - 0080 0000 0114 0004 0000 0002 0123 0008 [.....#..]
0050 - 0000 0000 0040 0000 0124 0004 0000 0001 [.....@...$......]
Exit with code: 0
```

Following is an example of an output from `inqj` command:

```
[root@franc ITDT]# ./itdt -f 4 0 8 0 inqj
{
 "command": "inquiry",
 "parameter": {
 "pageCode": 0,
 "evpd": false
 }
}
```

```

 },
 "length": 96 ,
 "data": {
 "version": "06" ,
 "vendorIdentification": "IBM" ,
 "productIdentification": "03592E07" ,
 "productRevisionLevel": "3C84"
 }
}

```

Supported platforms: All

---

## logpage|logpagej

(Deprecated: **-o log**) This subcommand issues the **SCSI Log Sense** command to the device for the specified page and displays the log sense data. The generated output is in hexadecimal format. `logpagej` command is similar to **logpage** command but generates the output in JSON format.

Parameters:

**Page** `Logpage` (in hex without leading x)  
**[Subpage]** `Subpage` (in hex without leading x)

Following is an example of an output from `logpage` command:

```

[root@franc ITDT]# ./itdt -f 4 0 8 0 logpage 2
Issuing log sense for page 0x02...
Log Sense Page 0x02, Length 30
 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000 - 0200 001A 0002 6002 0000 0003 6002 0000 [.....`.....]
0010 - 0005 6004 0000 0000 0006 6002 0000 [..`.....`....]
Exit with code: 0

```

Following is an example of an output from `logpagej` command:

```

[root@franc ITDT]# ./itdt -f 4 0 8 0 logpagej 2
{
 "command": "log sense" ,
 "parameter": {
 "pageCode": 2 ,
 "subPage": 0
 } ,
 "length": 255 ,
 "data": {
 "subPage": "00" ,
 "pageLength": 26 ,
 "values": [
 {
 "paramCode": "Not supported" ,
 "paramId": "0002" ,
 "paramValue": "0"
 } ,
 {
 "paramCode": "Total Corrected Write Errors" ,
 "paramId": "0003" ,
 "paramValue": "0"
 } ,
 {
 "paramCode": "Total Write Kibibytes Processed" ,
 "paramId": "0005" ,
 "paramValue": "0"
 } ,
 {
 "paramCode": "Total Uncorrected Write Errors" ,
 "paramId": "0006" ,
 "paramValue": "0"
 }
]
 }
}

```

Supported platforms: All

---

## loop

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

Parameters:

**loop** `[Count]`

Supported platforms: All

---

## modepage



(Deprecated: **-o mod**) This subcommand issues the SCSI Mode Sense command to the device for the specified page and displays the mode sense data.

Parameters:

**modepage** **Page** (page in hex without leading x)

Supported platforms: All

---

## prevent

(Deprecated: **-o lck, lock**) Prevent medium removal for tape or changer devices (lock door). The counter command for this command is **allow**.

Parameters:

**None**

Supported platforms: All

---

## print

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

Parameters:

**print** **Text**

Supported platforms: All

---

## qrypath

(Deprecated: **-o phs, path, checkpath**) 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.

Parameters:

**None**

Note: ITDT shows the entire path information for all the commands.

Supported platforms: All

---

## qryversion

(Deprecated: **-o drv**) This subcommand prints out the current version of the IBM® device driver.

Parameters:

**None**

Supported platforms: All

---

## release

(Deprecated: **-o rel**) This subcommand explicitly releases a device and makes it available for other hosts by issuing the SCSI Release command.

Parameters:

**None**

Supported platforms: All

---

## reqsense

(Deprecated: **-o req**) This subcommand issues the SCSI Request Sense command to the device and displays the sense data in hex format.

Parameters:

**None**

Supported platforms: All

---

## reserve

(Deprecated: **-o res**) This subcommand explicitly reserves a device by issuing the SCSI Reserve command.

Parameters:

**None**

Supported platforms: All

---

## scan|scanj

**scan** command scans the system for connected tape and changer devices, and displays the list of connected devices. For each detected device following information will be reported. **scanj** command is similar to **scan** command but generates the output in JSON format.

- Special file name
- Vendor ID
- Firmware version
- Serial-number
- SCSI bus address
- Associated changer
- Device driver name

Parameters:

```
[-force-generic-dd]
[-o formatstring]
[-start=device]
[-end=device]
[-exclude=device]
[-showallpaths]
```

Following is an example of an output from **scan** command:

```
[root@franc ITDT]# ./itdt scan
Scanning SCSI Bus ...
#0 /dev/IBMtape7n - [03592E07]-[3C84] S/N:000013D00123 H4-B0-T8-L0 (IBM-Device)
#1 /dev/IBMtape6n - [03592E07]-[3C84] S/N:000013D00123 H4-B0-T7-L0 (IBM-Device)
#2 /dev/IBMtape5n - [03592E05]-[1F05] S/N:000001365271 H4-B0-T6-L0 (IBM-Device)
Exit with code: 0
```

Following is an example of an output from **scanj** command:

```
[root@franc ITDT]# ./itdt scanj
[
 {
 "index": 0 ,
 "driverName": "/dev/IBMtape7n" ,
 "modelName": "03592E07" ,
 "firmwareVersion": "3C84" ,
 "serialNumber": "000013D00123" ,
 "hostAdapterId": 4 ,
 "busNumber": 0 ,
 "targetId": 8 ,
 "lunId": 0 ,
 "deviceSpecialFile": "IBM"
 } ,
 {
 "index": 1 ,
 "driverName": "/dev/IBMtape6n" ,
 "modelName": "03592E07" ,
 "firmwareVersion": "3C84" ,
 "serialNumber": "000013D00123" ,
 "hostAdapterId": 4 ,
 "busNumber": 0 ,
 "targetId": 7 ,
 "lunId": 0 ,
 "deviceSpecialFile": "IBM"
 } ,
 {
 "index": 2 ,
 "driverName": "/dev/IBMtape5n" ,
 "modelName": "03592E05" ,
 "firmwareVersion": "1F05" ,
 "serialNumber": "000001365271" ,
 "hostAdapterId": 4 ,
 "busNumber": 0 ,
 "targetId": 6 ,
 "lunId": 0 ,
 "deviceSpecialFile": "IBM"
 }
]
```

By using the optional parameter **-force-generic-dd**, the usage of the generic Operating System driver (not with the IBM® Tape Device Driver) is forced.

With the parameters `-start`, `-end`, and `-exclude`, the scan can be limited to only a particular HBA or target ID range (partial scan). It starts at `-start=device` to search for supported devices until `-end=device` is reached. Devices that are specified with `-exclude` are skipped during the scan. Multiple excluded devices must be separated by colons. Supported formats are devicename and H B T L.

The format string controls the output and specifies how the connected devices must be reported. It can include any alphanumeric character. The default format string is `"### %D - [%P] - [%F] S/N: %S H%H-B%B-T%T-L%L"`.

The following list contains all the interpreted identifiers:

```
%D device name
%V vendor name
%P product name
%F firmware version
%S serial number
%H host adapter
%B bus number
%T target id
%L logical unit
%h host adapter as hexadecimal value
%b bus number as hexadecimal value
%t target id as hexadecimal value
%l logical unit as hexadecimal value
%I interface type
%# enumeration number
%C serial number of associated changer device
```

Any combination of the identifiers that are listed here are supported.

An integer that is placed between a % sign and the format command acts as a minimum field width specifier. A negative value uses right text alignment.

Following is an example of `scan` command with format string:

```
./itdt scan -o "%-2#. %-18S %-12P %F %I"
Scanning SCSI Bus ...
0. 1310115166 ULT3580-TD4 97F2 Fibre
1. 1310034311 ULT3580-TD4 A232 Fibre
2. 1168001104 ULT3580-TD5 B170 Fibre
3. 00L4U78D6118_LL0 3573-TL 9.20 N/A
4. 00L2U78G8705_LL0 3573-TL 9.20 N/A
5. 000001327208 03592E06 26CA Fibre
6. cigen22164 ULT3580-TD2 73V1 Fibre
7. 000001365271 03592E05 1DD1 Fibre
```

Exit with code: 0

The option `-showallpaths` forces ITDT to display all available data paths that are detected during the device scan. The default device serial number-based filtering is not performed.

Supported platforms: All

---

## sleep

Sleep for the specified number of seconds before running the next subcommand.

Parameters:

**sleep** [Seconds]

Supported platforms: All

---

## tur

(Deprecated: **-o tur**) This subcommand issues the SCSI Test Unit Ready command to the device.

Parameters:

**None**

Supported platforms: All

---

## vpd

This subcommand displays Vital Product Data (VPD) that are part of the Inquiry command data and outputs Manufacturer, Product Identification and Revision Level.

Parameters:

**None**

Supported platforms: All

---

## append

Opens the device in append mode. The file access permission is Write Only.

Parameters:

**None**

Supported platforms: All, but on Windows this open mode is not supported by the IBM® Tape Device Driver. On HP-UX this open mode is remapped to r/w by the IBM Tape Device Driver.

---

## bsf

(Deprecated: **-o bsf**) 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.

Parameters:

**bsf [Count]**

Supported platforms: All

---

## bsr

(Deprecated: **-o bsr**) This subcommand backward spaces Count records. An optional count can be specified. The default is 1.

Parameters:

**bsr [Count]**

Supported platforms: All

---

## channelcalibration

This subcommand issues a self diagnostic command for drive servo channel calibration. The channel calibration returns PASSED or FAILED regarding the result of the command.

Parameters:

**None**

Supported platforms: All

---

## chgpart

This subcommand changes the current active tape partition to a new partition specified by Number. Optionally, a Blockid can also be specified. If Blockid is omitted, the tape is positioned at the start of the new partition. Otherwise, the tape is positioned at the Blockid specified.

Parameters:

**chgpart Number [Blockid]**

Supported platforms: All

---

## density

(Deprecated: **-o gdn / -o rds**) 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.

Parameters:

**None**

Note: ITDT-SE outputs detailed information on all platforms.

Supported platforms: All

---

## display

(Deprecated: **-o msg**) This subcommand displays a message on the display panel of the tape device. Up to 16 characters can be used for the message. If the message is longer than eight characters, the display alternates between the first eight characters and the remainder of the message.

Parameters:

```
display "message1" "message2"
```

Supported platforms: All

---

## erase

(Deprecated: **-o era**) This subcommand writes EOD at the current position and erases the tape from EOD to the end of current partition.

Parameters:

**None**

Supported platforms: All except i5/OS operating system

Note: The **erase** command triggers a long erase of the cartridge that sets EOD to the current position. Then, it writes the Data Set Separator (DSS) pattern from the new EOD to the end of the current partition. This process overwrites any data that is on the cartridge after the current logical position. To remove the entire cartridge, the user must remove all partitions (use the **rmp** command for LTO 5, TS1140, and newer drives). Then, issue the **rewind** command before the **erase** command.

Examples:

For LTO 5 / TS1140 and later

```
itdt -f <device name> load rmp rewind erase
```

For all earlier LTO and Enterprise drive generations

```
itdt -f <device name> load rewind erase
```

---

## formattape

This command formats the media on the device.

Parameters:

```
[-immed] This parameter immediately returns the status of the device.
[-verify] This parameter helps the device to run a verification after media is formatted.
[-mode] Format type
```

Supported platforms: All

---

## fdp

This subcommand creates fdp (fixed data partitions) wrap wise. The default size for LTO 5 of partition 0 in this case is 1425 GB and the size of partition 1 is 37.5 GB. It also works for TS1140 but the size depends on the used cartridge. Supported by LTO 5, TS1140, and later.

Parameters:

**None**

Supported platforms: All

---

## fdpl

This subcommand creates fdp (fixed data partitions) longitudinal. The command is valid only for TS1140 and later drives and creates partitions 0 and 1 on the cartridge. The size depends on the used cartridge.

Parameters:

**None**

Supported platforms: All

---

## fsf

(Deprecated: **-o fsf**) 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.

Parameters:

**fsf** [Count]

Supported platforms: All

---

## fsr

(Deprecated: **-o fsr**) This subcommand forward spaces count records. An optional count can be specified. The default is 1.

Parameters:

**fsr** [Count]

Supported platforms: All

---

## getparms

(Deprecated: **-o parms / status / -o gpa**) Get and show drive, media and driver parameters.

Parameters:

**None**

Supported platforms: All

---

## idp

This subcommand creates initiator defined partitions (IDP) wrap wise on tape. The parameter pSize0 is used to specify the size of partition 0 and the parameter pSize1 is used to specify the size of partition 1. One of pSize0 or pSize1 must have a value that is entered in hex matching 37.5 \* n with (1 <= n <= 38) to specify the wanted size of that partition. The other parameter of pSize0 or pSize1 must have the value 0xFFFF to specify that the remaining capacity is used for that partition. If 0xFFFF is not used for one of the parameters, pSize0 or pSize1, the drive might reject the command | unless pSize0 and pSize1 exactly match predefined allowable values.

For TS1140 and later drives (not LTO), the parameters pSize2 and pSize3 are valid and they follow the same rules as described earlier.

For example: If you want a 37.5 GB partition (the minimum size partition) in partition 0 and the remainder in partition 1, then set pSize 0 to 0x26 and pSize1 to 0xFFFF. This action results in a volume with a 37.5 GB sized partition 0 and a 1425 GB sized partition 1.

Parameters:

```
idp pSize0 pSize1
pSize0: size of partition 0
pSize1: size of partition 1
```

Example Call:

```
idp 0x26 0xffff
```

Supported platforms: All

---

## idpl

This subcommand creates initiator defined partitions (IDP) longitudinal wise on tape. The parameter pSize0 is used to specify the size of partition 0 and the parameter pSize1 is used to specify the size of partition 1. One of pSize0 or pSize1 must have a value that is entered in hex matching 37.5 \* n with (1 <= n <= 38) to specify the wanted size of that partition. The other parameter of pSize0 or pSize1 must have the value 0xFFFF to specify that the remaining capacity is used for that partition. If 0xFFFF is not used for one of the parameters, pSize0 or pSize1, the drive might reject the command | unless pSize0 and pSize1 exactly match predefined allowable values.

For TS1140 and later drives (not LTO), the parameters pSize2 and pSize3 are valid and they follow the same rules as described earlier.

For example: If you want a 37.5 GB partition (the minimum size partition) in partition 0 and the remainder in partition 1, then set pSize 0 to 0x26 and pSize1 to 0xFFFF. This action results in a volume with a 37.5 GB sized partition 0 and a 1425 GB sized partition 1.

Parameters:

```
idp pSize0 pSize1
pSize0: size of partition 0
pSize1: size of partition 1
```

Example Call:

```
idp 0x26 0xffff
```

Supported platforms: All

---

## 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 that is generated from this subcommand can be large depending on the amount of data on the tape and must usually be redirected to a file.

Parameters:

```
list [-start blockid] [-count recordcount]
```

Supported platforms: All

---

## load

(Deprecated: **-o lod**) This subcommand issues a **SCSI Load** command to load a tape. The subcommand loops all subsequent subcommands continuously or a number of times if the Count parameter is specified. Also refer to the sleep subcommand.

Parameters:

```
load [-amu 0|1]
```

amu enables or disables archive mode unthread during **Load** command.

Supported platforms: All

---

## logsense

Retrieves all Log Sense pages and outputs them as hex.

Parameters:

```
None
```

Supported platforms: All

---

## qrypar/qrypart

Queries and displays tape partitioning information.

Parameters:

```
None
```

Supported platforms: All

---

## qrylbp

Queries and displays logical block protection information.

Parameters:

```
None
```

Supported platforms: All

---

## qrypos

(Deprecated: **-o gpo**) This subcommand displays the current tape position.

Parameters:

```
None
```

Supported platforms: All

---

## qrytcpip

This subcommand outputs the current drive TCP/IP configuration. Only supported with LTO 5, TS1140, and later drives. Outputs adapter and TCP/IP address information for IPv4 and IPv6 with address, port and subnet mask. For example:

```
./itdt -f /dev/IBMtape3 qrytcpip
Reading current TCP/IP Configuration...
Number of Port Descriptors 1
Port Descriptor for Port 1
```

```
Number of Socket Descriptors 2
Adapter:1 IPV4 9.11.22.111/23 DHCP:0
Adapter:2 IPV6[0002:0000:0000:0000:0007:0008:0000:0000]/24
Active IP Addresses:
IPv4: 169.254.0.3
Exit with code: 0
```

Parameters:

**None**

Supported platforms: All

---

## qrytemp

Reads the thermal sensor values of a tape drive and writes the following output:

```
Thermal Value, Max Thermal Value,
Fencing Threshold and Fencing Removal Threshold.
All values are units of °C.
```

Parameters:

**None**

---

## read

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 that are 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 tape are read.

Parameters:

```
read -d Dest [-c Count]
```

Supported platforms: All

---

## readattr

This subcommand reads a Medium Auxiliary Memory (MAM) attribute from a cartridge to the destination file name that is specified with the -d flag. The partition parameter -p is a number in the range 0 - 3. The attribute parameter -a is the hexadecimal value of the identifier. All three parameters are required.

Parameters:

```
readattr -p[0|1|2|3] -a Identifier -d DestinationPathFile
```

Supported platforms: All

---

## resetdrive

This subcommand issues a Send Diagnostic command (Reset Drive subcommand) to reset the device.

Parameters:

**None**

Supported platforms: All

---

## rmp

Remove partitioning.

Parameters:

**None**

Supported platforms: All

---

## runtimeinfo/qryruntimeinfo



This subcommand is used to query the dynamic runtime information. Dynamic runtime information allows an initiator to set dynamic runtime attributes (DRA) about itself into a device server. The device server then associates those attributes to the I\_T\_L nexus and uses the information and associations for enhanced data collection and debugging.

Parameters:

**None**

Supported platforms: All (supported since LTO 5 and 3592 E07)

---

## rewind

(Deprecated: **-o rew**) Rewinds the tape.

Parameters:

**None**

Supported platforms: All

---

## rtest

(Deprecated: **-o rea**) This subcommand runs 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, then the count specifies the number of blocks to read on each repetition. If the block size is zero (variable), then 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.

Parameters:

```
read [-b Blocksize] [-c Count] [-r Repetition]
```

Supported platforms: All

---

## rwtest

This subcommand runs 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, then the count specifies the number of blocks to write on each repetition. A single transfer transmits (block size \* count) bytes. The operation is rejected if the total amount exceeds the transfer size the system is capable of. If the block size is zero (variable), then the count specifies the number of bytes to write on each repetition. The default is a block size of 10240 bytes, a count of 20 blocks, and a repetition of 1.

In SCSI Generic mode, the actual transfer size equals to *blocksize*. In IBM Device Driver mode, the transfer size is *blocksize \* blocks per read/write*. In either case, the blocksize must be equal to or less than the maximum Host Bus Adapter supported transfer size. The maximum supported transfer size for SCSI Generic is 1048576 Bytes and for IBM Device Driver is 500 MB.

Parameters:

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

Supported platforms: All

---

## sdp

Creates SDP (Select Data Partitions) wrap wise on tape.

Parameters:

```
sdp count
For LTO (5 and higher) only the values 0 and 1 are valid.
Using value 0 as parameter leads to partition 0 with 1.5TB
and partition 1 does not exist.
Using value 1 as parameter leads to partition 0 with 750 GB
and partition 1 with 712.5 GB.
For TS1140 and later drives values 0, 1, 2 and 3 are valid.
Using value 0 as parameter will create only one partition,
value 1 creates two and so on.
The sizes of the partitions are depending on the cartridge used in drive.
```

Supported platforms: All

---

## sdpl

Creates SDP (Select Data Partitions) longitudinal on tape.

Parameters:

**sdpl count**  
For TS1140 and later drives values 0 and 1 are valid.  
Using the value 0 as the parameter creates partition 0 only  
and the value 1 as the parameter creates partitions 0 and 1.  
The sizes of the partitions depends on the cartridge  
used in drive.

Supported platforms: All

---

## seed

(Deprecated: **-o eod**) Spaces to end of data on the tape.

Parameters:

**None**

Supported platforms: All

---

## setparm

(Deprecated: **-o spa / valid / compress / nocompress / sili / nosili / autoload / noautoload / retain / noretain**)

ITDT-SE uses the new setparm option that corresponds to the current interactive mode options.

Parameters: The Value is

- 0-65535 for the blocksize parameter
- 0-100 for the capacity parameter (=percentage)
- 1 (SCSI) and 2 (AIX) for recordspacemode
- The Volume Id string for the valid parameter
- NONE|ASSO|PERS|WORM for the writeprotect parameter
- 0 for off/no and 1 for on/yes for setparm autoload, autodump, buffering, compression, immediate, readpastfilemark, sili, simmim, trace, trailer, volumelogging

Supported platforms: All, but only a subset of the parameters is supported by the platform's device drivers.

Table 1. Supported platforms

|                                   | Linux | Windows        | AIX | Solaris | HP-UX |
|-----------------------------------|-------|----------------|-----|---------|-------|
| setparm autoload                  |       |                | X   |         |       |
| setparm autodump                  | X     |                |     |         |       |
| setparm blocksize                 | X     | X              | X   | X       | X     |
| setparm buffering                 | X     |                | X   | X       | X     |
| setparm capacity                  | X     | X <sup>3</sup> | X   | X       | X     |
| setparm compression               | X     | X              | X   | X       | X     |
| setparm datasafemode <sup>1</sup> | X     | X              | X   | X       | X     |
| setparm immediate                 | X     |                | X   | X       | X     |
| setparm readpastfilemark          | X     |                |     |         |       |
| setparm recordspacemode           |       |                | X   |         |       |
| setparm sili                      |       |                | X   | X       | X     |
| setparm simmim                    | X     |                |     |         |       |
| setparm skipsync <sup>2</sup>     | X     | X              | X   | X       | X     |
| setparm sleepmode                 | X     | X              | X   | X       | X     |
| setparm trace                     | X     |                |     |         |       |
| setparm trailer                   | X     |                | X   | X       | X     |
| setparm valid                     |       |                | X   |         |       |
| setparm volumelogging             | X     |                | X   |         |       |
| setparm writeprotect <sup>2</sup> | X     | X <sup>3</sup> | X   | X       | X     |
| setparm archivemodeunthread       | X     | X              | X   | X       | X     |

Note:

1. The datasafemode can be set to YES or NO when no cartridge is loaded. When a cartridge is loaded, the datasafemode can be set only to YES.
2. Depending on the support of the device.
3. Only supported by 3592.

---

## setpos

(Deprecated: **-o spo / asf**) 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 by using the **qrypos** subcommand.

Parameters:

**setpos** [Blockid]

Supported platforms: All

---

## settcpip

This subcommand sets the drive ethernet port TCP/IP settings for LTO 5, TS1140, and later drives. Either a static IPv4 or IPv6 address can be set or DHCP enabled.

Example DHCP:

```
./itdt -f /dev/IBMtape3 settcpip DHCP
Initializing device...
Setting TCP/TP Configuration...
Reading current TCP/TP Configuration...
Number of Port Descriptors 1
Port Descriptor for Port 1
Number of Socket Descriptors 2
Adapter:1 IPV4 9.11.22.111/23 DHCP:1
Adapter:2 IPV6 [0002:0000:0000:0000:0007:0008:0000:0000]/24
Active IP Addresses:
Exit with code: 0
```

IPv4 or IPv6 addresses are entered in the syntax a.b.c.d:/subnet\_mask\_length where a, b, c, and d are values with 1 to 3 digits. If the optional parameter subnet\_mask\_length is not specified, the current setting is kept.

Example static IPv4:

```
./itdt -f /dev/IBMtape3 settcpip 9.155.27.14/23
Initializing device...
Setting TCP/TP Configuration...
Reading current TCP/TP Configuration...
Number of Port Descriptors 1
Port Descriptor for Port 1
Number of Socket Descriptors 2
Adapter:1 IPV4 9.155.27.14/23 DHCP:0
Adapter:2 IPV6 [0002:0000:0000:0000:0007:0008:0000:0000]/24

Active IP Addresses: IPV4:

169.254.0.3 IPV4:

9.155.27.46 IPV4:

9.155.27.14

Exit with code: 0
```

Parameters:

**dhcp**, **address**[/subnet\_mask\_length]

Supported platforms: All

Note: With the current firmware level, the device can be reached (ping, FTP) only within the same subnet. For example, the sample is configured with a static IP address (9.155.27.14) and DHCP enabled. The drive can be pinged only within the same subnet (9.155.27.xxx). Both IP addresses (9.155.27.14 and the DCP address 9.155.27.46) are active and can be used.

---

## sync

(Deprecated: **-o syn**) This subcommand synchronizes buffers/flushes the tape buffers to tape.

Parameters:

**None**

Supported platforms: All

---

## unload

(Deprecated: **-o off / offline / rewoffl**) This subcommand rewinds and unloads the tape.

Parameters:

**unload** [-amu 0|1]

amu enables or disables archive mode unthread during **Load** command.

Supported platforms: All

---

## verlbp

This subcommand verifies logical block protection written blocks. The verification length can be set with parameter value filemarks count or with EOD.

Parameters:

**filemarks** (numeric value) | **eod**

Supported platforms: All

---

## weof

(Deprecated: **-o eof / eof**) These subcommands write count filemarks. An optional count can be specified. The default is 1.

Parameters:

**weof** [Count]

Note: The weof parameter [count] is optional, if it is not supplied, one filemark is written.

Supported platforms: All

---

## write

This subcommand writes the source file specified with the -s flag on the tape. In case the parameter 'raw' is specified, the blocksize specified in setparm (setparm blocksize) is used instead of the default blocksize of 64 kB. The parameter count specifies the amount of blocks which should be written.

Parameters:

**write** [-raw] [-count] -s Source

Supported platforms: All

---

## writeattr

This subcommand writes a Medium Auxiliary Memory (MAM) attribute from a cartridge from the source file name that is specified with the -s flag. The partition parameter -p is a number in the range 0 - 3. The attribute parameter -a is the hexadecimal value of the identifier. All three parameters are required.

Parameters:

**writeattr** -p[0|1|2|3] -aIdentifier -sSourcePathFile

Supported platforms: All

---

## wtest

(Deprecated: **-o wri**) This subcommand runs 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.

Parameters:

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

Supported platforms: All

---

## audit

(Deprecated: **-o aud / -o ier**) 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.

Parameters:

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

Supported platforms: All

---

## cartridge location

This subcommand with no parameters issues the SCSI Read Element Status command to the device to report all slots with the cartridge location information. Using the optional parameters Slot and Count issues the SCSI Read Element Status to the device for a specific starting Slot address and optionally the Count specifies the number of slots to return. If Count is omitted, it defaults to 1.

Parameters:

**cartridgelocation** [Slot [Count]]

Supported platforms: All

---

## elementinfo

(Deprecated: **-o ele**) This subcommand displays element information (number and address) of each element type.

Parameters:

**None**

Supported platforms: All

---

## exchange

(Deprecated: **-o exh**) This subcommand issues the SCSI Exchange Medium command to the device by using the Source, Dest1, and Dest2 addresses specified. This command runs the equivalent function of two **Move Medium** commands. The first moves the cartridge from the element address that is specified by the Dest1 parameter to the element address specified by the Dest2 parameter. The second moves the cartridge from the element address that is specified by the Source parameter to the element address specified by the Dest1 parameter.

Parameters:

**exchange** Source Dest1 Dest2

Supported platforms: All

---

## inventory

(Deprecated: **-o inv**) 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 is returned. If the optional -v parameter is used, then only the element status information for the specified Valid if found is returned.

Parameters:

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

Note: ITDT supports the optional parameters on all platforms.

Supported platforms: All

---

## move

(Deprecated: **-o mov**) This subcommand issues the SCSI Move Medium command by using the source and destination addresses specified. The element addresses can be obtained with the **elementinfo** subcommand.

Parameters:

**move** Source Dest

Supported platforms: All

---

## position

(Deprecated: **-o pos**) This subcommand issues the SCSI Position to Element command by using the destination specified.

Parameters:

**position** Dest

Supported platforms: All

---

## dump

(Deprecated: **-o sdp**) This subcommand forces a dump of the tape drive and stores the dumps before and after the force dump in the ITDT-SE output folder with the ITDT-SE naming convention (serialnumber.a.gz and serialnumber.b.gz).

Parameters:

**None**

Supported platforms: All

---

## ekmtest

Test encryption key path/setup.

Parameters:

**None**

Supported platforms: AIX®, Linux®, Solaris, Windows

---

## encryption

Query tape drive encryption settings and display the encryption state.

Parameters:

**None**

Supported platforms: All

---

## ucode

(Deprecated: **-o dmc**) This subcommand downloads microcode to the device. The Filename is a file that contains the ucode.

The TS4300 (3555) tape library does not support code update with SCSI commands. ITDT integrated RESTful API functions to support code update by using the HTTPS connection of the library. The process is similar to the regular code update sequence, but in addition the user must enter credentials during the code update sequence.

Parameters:

```
ucode [-password loginpassword -user username -ipaddress library Address] Filename
```

Supported platforms: All

---

## tapephcp

Creates a physical copy of a tape cartridge. The created cartridge has the same physical layout and contents as the origin cartridge. The amount of transferred data and the current data transfer rate is displayed every 3-5 minutes. Tapephcp is supported for LTO and 3592 tape drives and can therefore be used for data migration. A **tapephcp** command that is issued to copy data from a 3592 drive to an LTO drive or from an LTO Gen 4 to an LTO Gen 3 works, if the amount of used data from the source device is equal or less than the capacity of the destination device.

Parameters:

```
tapephcp [-cqs Memorysize] source destination
```

Source and destination can either a special device file name, or a tape image file name. The special device file format is identical to the format specified in chapter 3.19 parameter “-f device”.

Examples: Tape to Tape copy with the IBM® tape driver

```
./itdt tapephcp /dev/IBMtape0 /dev/IBMtape1
Tape to Image File using generic interface
./itdt tapephcp 3 2 1 0 /tmp/MyTapeImage.img
Image File to Tape using IBM Tape driver
./itdt tapephcp /tmp/MyTapeImage.img /dev/IBMtape1
Tape to Tape copy with adjusted memory allocation
./itdt tapephcp -cqs 1000 /dev/IBMtape0 /dev/IBMtape1
```

To ensure maximum performance for tape to tape copy actions, tapephcp allocates a read buffer of 2500 MB (assuming the maximum system block size is 1 MB). If the system does not provide this buffer, the operation is aborted with **MEMORY ALLOCATION FAILED**. The value for memory allocation can be changed with an integer value for the parameter cqs (Command Queue Size). The cqs value must be multiplied with the maximum supported system blocksize to determine the size of memory that is allocated by ITDT. Supported copy operations are from tape to tape, from image file to tape, and from tape to image file. On UNIX operating systems, a warning to check the file size limit is shown before an image file is created, if the maximum file size is below 2 GB.

---

## ltfsphcp

Creates a physical copy of an LTFS formatted cartridge. `ltfsphcp` is based on `tapephcp`. The LTFS specified parameters `volumeuuid` and `VCI` are adjusted during this copy operation. The created cartridge has the same physical layout as the origin cartridge. Expect the `volumeuuid` to be identical to the contents of the two cartridges. The amount of transferred data and the current data transfer rate is displayed every 3-5 minutes. When `ltfsphcp` is used with a non-LTFS formatted cartridge, the behavior of `ltfsphcp` is identical to `tapephcp`.

Parameters:

```
ltfsphcp [-cqs Memorysize] [-VOL1 volume id] source destination
```

Source and destination can either a special device file name, or a tape image file name. The special device file format is identical to the format specified in chapter 3.19 parameter “-f device”.

Examples:

```
Tape to Tape copy using IBM tape driver
./itdt ltfsphcp /dev/IBMtape0 /dev/IBMtape1
Tape to Image File using generic interface
./itdt ltfsphcp 3 2 1 0 /tmp/MyTapeImage.img
Image File to Tape using IBM Tape driver
./itdt ltfsphcp /tmp/MyTapeImage.img /dev/IBMtape1
Tape to Tape copy with adjusted memory allocation
./itdt ltfsphcp -cqs 1000 /dev/IBMtape0 /dev/IBMtape1
```

To ensure maximum performance for tape to tape copy actions, `ltfsphcp` allocates a read buffer of 2500 MB (assuming the maximum system block size is 1 MB). If the system does not provide this buffer, the operation is aborted with `MEMORY ALLOCATION FAILED`. The value for memory allocation can be changed with an integer value for the parameter `cqs` (Command Queue Size). The `cqs` value must be multiplied with the maximum supported system blocksize to determine the size of memory that is allocated by ITDT. Supported copy operations are from tape to tape, from image file to tape, and from tape to image file. On UNIX operating systems, a warning to check the file size limit is shown before an image file is created, if the maximum file size is below 2 GB.

For LTFS formatted cartridges, the volume identifier can be changed for a physical target with the parameter `VOL1`. The value must be alphanumeric with a minimum of 1 and a maximum of 6 characters.

---

## verify

Verifies the physical contents of two cartridges. The physical data layout and the binary data are compared.

Parameters:

```
verify source destination
```

Source and destination can either a special device file name, or a tape image file name. The special device file format is identical to the format specified in chapter 3.19 parameter -f.

Examples:

```
./itdt verify /dev/IBMtape0 /dev/IBMtape1
```

---

## devicestatistics

This command collects relevant device parameters from several log pages and inquiry data and prints out in a table.

---

## checkltfsreadiness

This subcommand issues the LTFS Readiness Check test.

The LTFS Readiness Check analyzes the Operating System and tape drive environment to ensure that the IBM® Linear Tape file system can be installed. This test checks the Operating System version, the tape device driver version, the tape drive firmware, and the LTFS HBA requirements. LTFS Readiness Check requires an empty data cartridge.

Parameters:

None

---

## ltfsdefragmentation

On an LTFS formatted cartridge, the physical data records for a single file can be fragmented across the entire media. When such a file is accessed, a long response time might result. The tape drive must locate to different cartridge positions to retrieve the entire contents the file. If the first data records of a file are stored at the end of the tape and the other records are stored at the beginning of the media, the tape drive must run several times intensive seek operations to fulfill the complete file retrieval. This subcommand creates a copy of the cartridge with unfragmented content.

As an initial step, ITDT stores the complete content of the source tape media in a Tape Image file that is on an HDD. Using this Tape Image file and the ITDT image file backend driver for LTFS, LTFS is able to mount the previously created Tape Image file as a read-only volume. As the final step, the data is defragmented by copying the files from the mounted Tape Image file to the mounted destination cartridge. This algorithm avoids any seek operations on the physical tape device. The seek operations are completed on the temporary ITDT image file that is on a hard disk. The source and destination tape device are accessed at maximum media speed. The defragmentation of a cartridge can take up to 6 hours.

Parameters:

```
ltfsdefragmentation source tempdirectory destination [options]
```

Supported platforms: Linux x86\_64 only

Prerequisites: IBM LTFS SDE Version 2.2 (Build 4700 or later) and sufficient free hard disk space for temporary Tape Image file

Example:

```
./itdt ltfsdegragmentation /dev/IBMtape0 /tmp/tapeimages
/dev/IBMtape1 -verbose -mkltfsoption=--force
```

---

## standardtest

The Test function (Scan Menu Command [T]) checks if the tape device is defective and outputs a pass/fail result. This test requires a loaded cartridge.

Parameters:

```
standardtest [-forcedataoverwrite]
```

Supported platforms: All

---

## fullwrite

Runs the ITDT Full Write Test (Scan Menu Command [F]).

The Full Write function writes the entire cartridge with a given block size either with compressible or incompressible data and output performance data. This test requires a loaded cartridge.

Parameters:

```
fullwrite [-b Blocksize] [-compressible|-incompressible] [-forcedataoverwrite]
```

Supported platforms: All

---

## systemtest

Runs the ITDT Systemtest (Scan Menu Command [Y]).

The System Test is a short test that runs the following actions.

- Reveals system performance bottlenecks. Compressible data throughput values can reveal bandwidth limitations that are caused by the system or cabling or HBA.
- Measures performance variations across the different block sizes to find the ideal block size for the system configuration.

This test requires a loaded cartridge.

Parameters:

```
systemtest [-forcedataoverwrite]
```

Supported platforms: All

---

## tapeusage

Displays the tapeusage information (Scan Menu Command [U]).

This test requires a loaded cartridge.

Parameters:

**None**

Supported platforms: All

---

## hdp discover

Discovers the HD-P system and presents the logical and physical HD-P environment. The logical map is an n x n representation of the libraries, where a “1” is the indicator for a connection and a “-1” of no connection. In case the command is called without any device files, all attached TS3500/TS4500 are used.

Parameters:

```
hdp discover [-d device-file]
```

Supported platforms: AIX®, Linux® on P, and X64

A mixed environment (LTO and 3592 devices) is not supported.



---

## hdp senderror

Issues a Test Call Home function. In case the command is called without any device file, all attached TS3500/TS4500 are used.

Parameters:

**hdp senderror** [-d device-file]

Supported platforms: AIX®, Linux® on P, and X64

---

## hdp show

Shows the extended HD-P Import Export Elements. In case the command is called without any device file, all attached TS3500/TS4500 are used.

Parameters:

**hdp show** [-d device-file]

Supported platforms: AIX®, Linux® on P, and X64

---

## TS4500 REST over SCSI

The TS4500 supports a SCSI in band method for sending REST API commands and receiving HTTP responses that use SCSI **Write Buffer** and **Read Buffer** commands. The method is called REST over SCSI or RoS, for short. ITDT handles the RoS calls.

Syntax:

**ros** | **rosraw** **GET** | **PATCH** | **POST** url-endpoint [options] [-o filename]

Parameters:

- Use 'ros' for prettified output, 'rosraw' for raw mode (no indents, line breaks, Unicode filtering)
- HTTP-Method: GET | PATCH | POST
- Url-endpoint is the RoS method '/v1/...'
- Options in brackets {} - string must be masked with single quotation marks ('')
- Optional file name where the JSON is stored.

For a detailed URL endpoint definition, refer to the TS4500 RESTful documentation:

<https://www.ibm.com/support/knowledgecenter/STQRQ9/com.ibm.storage.ts4500.doc/rest.html>.

Example:

```
$/itdt -f /dev/smc30 ros GET /v1/accessors HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 461
[
 {
 "location": "accessor Aa", "state": "onlineActive", "pivots": "13256",
 "barCodeScans": "109653", "travelX": "9475",
 "travelY": "11640",
 "getsGripper1": "4488",
 "putsGripper1": "4487",
 "getsGripper2": "4486",
 "putsGripper2": "4465"
 },
 {
 "location": "accessor Ab", "state": "onlineActive", "pivots": "3496",
 "barCodeScans": "61212", "travelX": "2845",
 "travelY": "2139",
 "getsGripper1": "160",
 "putsGripper1": "154",
 "getsGripper2": "163",
 "putsGripper2": "148"
 }
]
Exit with code: 0
```

---

## Endpoints with file transfers

The URL-endpoints

- **GET** /v1/logs/[filename]/export
- **GET** /v1/library/saveConfig

store the content in a file. The file name is printed at the end.

Examples:

```
./itdt -f /dev/smc30 ros POST /v1/tasks '{"type":"inventoryTier0and1","location":"library"}'
./itdt -f /dev/smc30 ros GET /v1/logs/TS4500_LOG_AA004_20191106143648.zip/export
./itdt -f /dev/smc30 ros GET '/v1/diagnosticCartridges/DG 0200L4'
```

Notes:

1. The IBM® Tape Device Driver is sensitive to not ready conditions that can prevent the usage of RoS commands. For this reason, the generic operating system driver is recommended with RoS. For details, see [Generic Operating System driver with ITDT-SE](#).
2. The Microsoft Windows command prompt does not support a single quotation mark (') to send a string to an application. Every special character must be escaped. Example:

```
ros POST /v1/tasks '{"type":"inventoryTier0and1","location":"library"}'
```

works on Linux®, but must be sent on Windows as:

```
ros POST /v1/tasks {"type\\":\\"inventoryTier0and1\\",\\"location\\":\\"library\\"}
```

---

## TS4300: RESTful API

The TS4300 REST API is a simple application planning interface (API) to manage the 3U scalable tape libraries remotely over an HTTPS interface. This API is requested and needed for manufacturing and for automated test and monitoring systems.

Syntax:

```
https-address {login-credentials} GET | PATCH | POST url-endpoint [JSON data]
```

Parameters:

- https-address
- Login credentials in JSON format
- HTTP-Method: GET | PATCH | POST
- url-endpoint
- [JSON data]

---

### https-address

The IP address of the library.

---

### Login credentials

The library differentiates the following security levels:

- Admin Security
- User Security
- Service Security

The user/password that is used to log in. Additionally, for some product variants in case of service level login the service password (service\_password) and the administrator password (password) must be sent. In case the administrator password is not required the 'service\_password' must be set through the normal password field. Parameter:

```
'{"username":"administrator",
 "password":"password"
 ["service_password":"servicepassword"]}'
```

---

### http-method

The method for a dedicated request: GET, PUT, POST

---

### url-endpoint

Valid URLs in combination with the http method. If the parameter is required, it can be passed with ? and &.

For a detailed URL endpoint definition, refer to the TS4300 RESTful documentation:

[https://www.ibm.com/support/knowledgecenter/STAKKZ/ts4300\\_kc/con\\_3U\\_REST\\_overview.html](https://www.ibm.com/support/knowledgecenter/STAKKZ/ts4300_kc/con_3U_REST_overview.html).

---

### JSON data

Some commands need extra data in JSON format like 'serialnumber change'.

Example:

```
sudo ./itdt https://1.2.3.4 '{"username":"service","password":"123456"}' GET /library/baseinfo
{
 BaseInfo {
 "SerialNumber": "1234567890123455",
 "MacAddress_1": "00:0e:11:1c:31:b5",
 "MacAddress_2": "00:0e:11:1c:31:b7",
 "Vendor": "IBM",
 "ProductID": "3573-TL",
 "BaseFWRevision": "1.3.0.0-D00",
 "BaseFWBuildDate": "07-02-2019",
 "ExpansionFWRevision": "0.30",
 "WWNodeName": "5000000000000000",
 "RoboticHWRevision": "4",
 "RoboticFWRevision": "0.13",
```

```

"RoboticSerialNumber": "12345678901234", "NoOfModules": "2",
"LibraryType": "32"
},
"ModulesInfo": [
{
"PhysicalNumber": "4",
"LogicalNumber": "1",
"ReadyStatus": "TRUE",
"SerialNumber": "3555L3A1234567"
},
{
"PhysicalNumber": "5",
"LogicalNumber": "2",
"ReadyStatus": "TRUE",
"SerialNumber": "3555E3A0234567"
}
]
}
Exit with code: 0

```

Notes:

1. ITDT sends a login request for each command.
2. Microsoft Windows command prompt does not support a single quotation mark (') to send a string to an application. That is, every special character must be escaped. The Microsoft Windows command shell is different and the command must be sent with the following form.

Example for Windows:

```

itdt.exe https://1.2.3.4 "{\"username\":\"service\",\"password\":\"123456\"}" GET /library/baseinfo
itdt.exe https://1.2.3.4 "{\"username\":\"service\",\"password\":\"123456\"}" GET "/library/getevents?
EventType=Info&MaxNum=2"

```

---

## Deprecated commands

The following is a list of commands that are currently available in this version of ITDT. However, in a future release the following commands and some alternate calls of the Common Command Scripting set are no longer available and the scripts that contain these commands must be changed. The scripts that use the deprecated commands must be changed for future editions.

General commands

- -o dis / disablepath
- -o ena / enablepath
- fuser
- kill
- passthru
- resetpath

Tape commands

- bsfm
- -o chk
- fsfm
- -o grs
- -o gmi
- qryinquiry
- qrysense
- -o ret
- setblk
- -o gds
- getrecsize
- setrecsize

Medium Changer subcommands

- -o dvc
- mount
- demount

Service Aid commands

- fmrtape
- -o fdp
- reset
- -o qmc

---

## Alternative mount/demount script - sample for Windows

The script (load.drive.cmd) issues a **mount/umount** command to the specified library. If the drive is empty, a cartridge is moved from the first available storage slot to the drive. If the drive contains a cartridge, the cartridge is moved to the previous storage location.

Requirements:

- Windows operating system
- IBM® tape library with one or more tape drives
- IBM tape device driver or Generic SCSI driver
- Path variable must include ITDT executable

Usage: load-drive.cmd drivename|hbt1 [drivenumber]  
 drivename is the device driver name assigned by the IBM device driver  
 hbt1 is the associated SCSI address h=host, b=bus, T=target id and l=lun  
 drivenumber, logical drive number, default is 1

Example for using the IBM device driver: load-drive.cmd \\.\Changer0

Example for using the SCSI Generic driver: load-drive.cmd  
 "2 0 3 1"

```
load-drive.cmd:
@ECHO OFF
IF [%1]==[] (
 echo.
 echo Usage load-drive.cmd drivename^|hbt1 [drivenumber]
 echo.
 echo drivename is the drivename assigned by the IBM device driver
 echo example:load-drive.cmd \\.\Changer0
 echo.
 echo hbt1 is the associated SCSI address h=host, b=bus, T=target id and l=lun
 echo example: load-drive.cmd "3 0 2 1"
 echo.
 goto :EOF
)

set driveNumber=1
IF not [%2]==[] set driveNumber=%2
echo Loading/Unload Cartridge Drive:%driveNumber%

itdt -f %~1 inventory >inventory.txt
set /a count=0
set action=load
set sourceAddress=
set destination=
set driveAddress=
set slotAddress=
FOR /F "tokens=1,2,3,4,5" %%G IN (inventory.txt) DO (
 call :checkline %%G %%H %%I %%J %%K
)
GOTO :moveMedium

:checkline
set name=%1 %2
if "%name%"=="Drive Address" (
 set /a count+=1
 set currentTag=%name%
 set driveAddress=%3
)
if "%name%"=="Media Present" (
rem echo full drive addr:%driveAddress% slot addr:%slotAddress%
action:%action% source:%sourceAddress% Tag:%currentTag%

 if "%count%"=="%driveNumber%" if "%currentTag%"=="Drive Address" if "%4%"=="Yes" (
 set action=unload
 set sourceAddress=%driveAddress%
)

 if "%count%"=="%driveNumber%" if "%currentTag%"=="Drive Address"
 if not "%4%"=="Yes" (
 set action=load
 set destination=%driveAddress%
)

 if "%sourceAddress%"==" " if "%currentTag%"=="Slot Address"
 if "%action%"=="load" if "%4%"=="Yes" (
 set sourceAddress=%slotAddress%
)
)

if "%name%"=="Source Element" (
 if "%count%"=="%driveNumber%" if "%currentTag%"=="Drive Address"
 if "%action%"=="unload" (
 set destination=%5
)
)

if "%name%"=="Slot Address" (
 set currentTag=%name%
 set slotAddress=%4
)

GOTO:EOF
:moveMedium
echo %action% move media from %sourceAddress% to destination %destination%
itdt -f %~1 move %sourceAddress% %destination%
del inventory.txt
GOTO:EOF
```

---

## Standard Edition scripting commands: known limitations and deviations

The scripting commands **idp**, **sdp**, **fdp**, and **qrypt** are currently only supported by LTO 5, TS1140, and later drives.

When scripting, one must be aware of the following general deviations to the legacy tapeutil scripting command set.

The Verbose Mode parameter is ignored for the ITDT-SE implementation; instead always the full information is printed.

For some operations, the sense data length is limited to 32 bytes - this length is required by the ITDT test sequences.

The list command does not work on Linux® variants where the reported maximum SCSI transfer size is larger than the size the system can actually transfer.

Because ITDT-SE opens the device read/write by default in scripting mode, the WORM mode cannot be deactivated in scripting mode. Use the Tapeutil interactive mode instead to deactivate the WORM mode.

Scripting mode command deviations to legacy tapeutil (tapeutil is not changed):

1. The **erg** (Erase Gap) command is not supported.
2. The **mtdevice** command is not supported.
3. The **tell** command is not supported.
4. The **seek** command is not supported.
5. The **format** command is not supported.
6. The **-o qsn** command is not supported.
7. **path/querypath/path / qrypath / checkpath / -o phs** - output: always show all paths.  
The command is valid only in combination with an IBM® device driver.
8. **devinfo** - different output (decoded on all platforms)
9. **inquiry** - different output on Linux (like AIX® in hex)
10. **vpd** - different behavior on Solaris (like AIX)
11. **modepage** - HP-UX and Solaris output deviations (like AIX)
12. **inventory** - additional parameters on AIX available on all platforms
13. **cartridgelocation** - AIX parameter deviation available on all platforms
14. **mediainfo** - different output --> decode on all platforms  
The command is valid only in combination with an IBM device driver.
15. **setpos** - logical versus physical position, only set logical position as on AIX
16. HP-UX: **-o gpo -t 1|2** --> parameter **-t1|2** (logical, physical) is not supported.
17. **density** - output all information on all platforms
18. **setparm** (new) - work around the inability to set all parameters on all platforms except the undocumented HP-UX release
19. **getparms** (new) - retrieve all parameters on all platforms, independent of flag set
20. **qrynquiry** - output the same as on AIX
21. **logsense** - does not output header with device/serial/valid, example:  

```
05/12/09 15:10:44 Device: ULT3580- S/N: 1300000206 Valid: UNKNOWN
```
22. **erase** - does not work on i5/OS because of operating system limitation.
23. Using **-w x** parameter without the necessary open leads to a core at least on Solaris.

---

## IBM® Tape Diagnostic Tool - Graphical Edition

- [Installing ITDT - Graphical Edition](#)
- [Graphical Edition - known issues and limitations](#)
- [ITDT-GE user interface description](#)
- [Graphical Edition - Scan menu commands](#)
- [Graphical Edition - Copy Services](#)
- [Graphical Edition - visualizing log files](#)
- [Graphical Edition - Tapeutil menu commands](#)

---

## Installing ITDT - Graphical Edition

- [Installing ITDT-GE on Windows operating systems](#)
- [Installing ITDT-GE on Linux operating systems](#)

---

## Installing ITDT-GE on Windows operating systems

To install ITDT-GE on Windows, download the executable file `install_ITDT_GE_<version>.exe` on a directory of your choice.

Double-click the downloaded file to start the installer application.

ITDT-GE installer automatically uninstalls any previous version before the current one is installed.

For the graphical user interface (GUI), a minimum screen resolution of 1024\*768 pixels is required.

The supported Microsoft Windows operating systems for ITDT-GE are:

- Microsoft Windows Server 2016 / 2012R2 (64-bit x64)

---

## Installing ITDT-GE on Linux operating systems

To install ITDT-GE on Linux, download the executable file `install_ITDT_GE_<version>.bin` to a directory of your choice.

`install_ITDT_GE_<version>.bin` must be run by a user with root access rights.

The ITDT-GE installer automatically uninstalls any previous version before the current one is installed.

For the graphical user interface (GUI) a minimum screen resolution of 1024\*768 pixels is required. The supported Linux® operating systems for ITDT-GE are

- Linux Distributions with Kernel 2.6, glibc 2.2.5 and later (64-bit x64).

---

## Graphical Edition - known issues and limitations

This section describes the known issues and limitations of ITDT-GE.

- [Linux operating systems](#)
- [Windows operating systems](#)
- [All supported operating systems](#)

---

## Linux operating systems

It is recommended to not operate Security Enhanced Linux® (SELinux) in enforcing mode while ITDT-GE is running.

On Red Hat Enterprise Linux 4 and 5 and SuSE Enterprise Linux 10, starting the online help might cause an error message `Problems opening link / Unable to open web browser on {0}`. Workarounds are to issue the commands

```
a) ln -s /usr/bin/firefox /usr/local/bin/mozilla
or
b) export MOZILLA_FIVE_HOME=/usr/lib/firefox<version>
```

Replace with the appropriate path to your installed Firefox version before ITDT-GE is started.

---

## Gnome desktop

If you are using the Gnome desktop, be sure to log in to the desktop session as root to use ITDT-GE to prevent device access issues.

---

## Linux GTK

ITDT-GE requires GTK3+ to run since version 9.4.0. On Linux systems with GTK2, ITDT can be installed or updated but does not start afterward.

---

## Windows operating systems

On Microsoft Windows systems where the maximum transfer size is limited to less than 64 kB, the Dump and Firmware update operations do not work.

---

## Performance issues

If you are using Adaptec SCSI Host Bus Adapters, ensure that you are using the latest Adaptec Host Bus Adapter Drivers instead of the drivers that are shipped with the Windows operating system.

---

## Devices disappear after firmware update

After a firmware update, devices might disappear.

Repeated Scan operations might help to rediscover the device.

---

## All supported operating systems

---

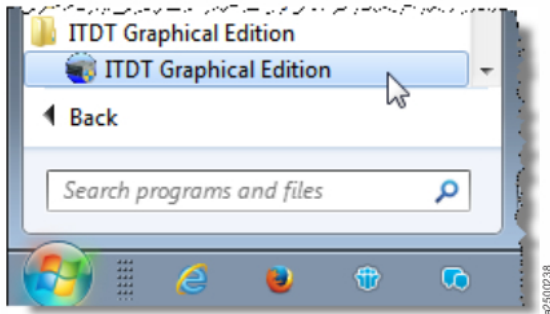
### Command timeout

There is no instant operation termination upon SCSI command timeout. For example, when the SCSI cable is unplugged after POST A is started.

When a command timeout condition occurs, ITDT might still continue to run more operations (like unmounting the cartridge) instead of instantly terminating with a timeout condition.

## ITDT-GE user interface description

Figure 1. Graphical Edition user interface



To start ITDT-GE on Windows, click the shortcut that is created by the installation process. On Linux®, no start menu entry is generated. Start ITDT-GE by opening a Terminal window, then switch to root user.

```
$ su -
```

Finally, start ITDT-GE with

```
$ /opt/ibm/itdt-ge/itdt-ge
```

The User Settings dialog box displays the first time that the program is run, allowing the entry of user specifications: User name, company name, output path, and log level.

The Output Path defines the location where the test reports and dumps are saved. The Windows default output path is

```
C:\Documents and Settings\<username>\.itdt-ge\output
```

or

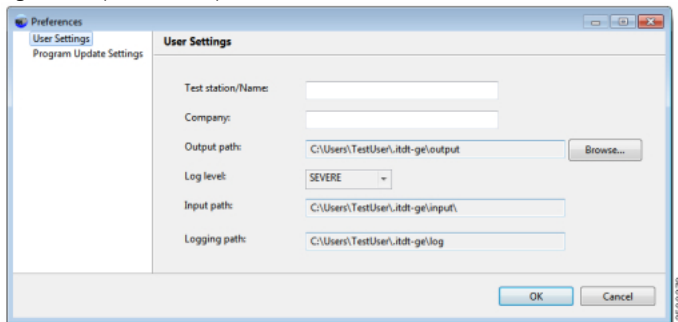
```
C:\Users\<username>\.itdt-ge\output
```

where <username> is the Windows user login name. The Linux default output path is

```
/root/.itdt-ge/output
```

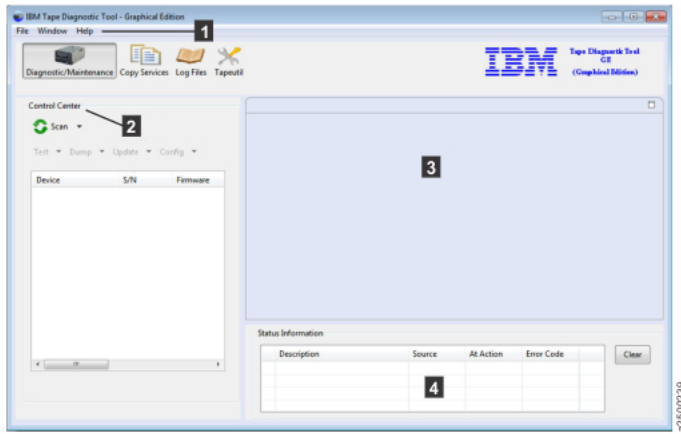
The Log Level must not be changed unless requested to do so by the IBM® Support Center. It is recommended that this information be provided to allow for further analysis of test results.

Figure 2. Graphical Edition preferences



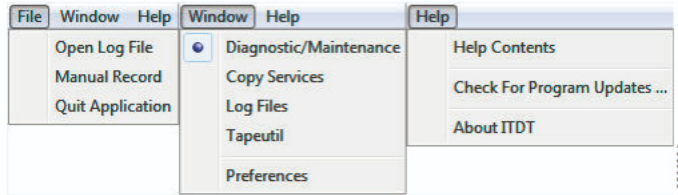
The ITDT-GE interface contains the following sections:

Figure 3. Graphical Edition interface



- Main menu ( 1 on [Figure 3](#))- in upper left (File, Window, Help) The following main program menu items are available:

Figure 4. Main program menu items

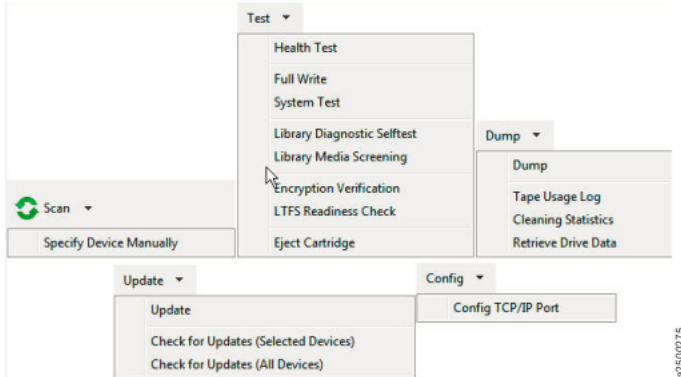


- Control Center ( 2 on [Figure 3](#)) - On left side (Device operations - Scan, Test, Dump, and Update) Extra device operations are available by using drop-down arrows.
- Test Lab ( 3 on [Figure 3](#)) - Located from center to right side (Contains running and previously run tests)
- Status Information ( 4 on [Figure 3](#)) - Located below the Test Lab (Contains summary results of tests)

The Control Center is the primary section the ITDT-GE operations.

The following toolbar buttons for the device operations are available.

Figure 5. Toolbar buttons



- Click the Scan menu item in the Control Center to display the list of tape devices found. When the scan is complete, select one device in the Test Lab by clicking the corresponding check box. Click the arrow next to the Scan menu item to add a device manually.
- Test  
Click the arrow next to the Test menu item to select an extended test.
- Dump  
Click the arrow next to the Dump menu item to select more log options.
- Update  
Click the Update menu item to start the firmware update. Click the arrow next to the Update menu item to select Online update options.
- Config  
Click the TCP/IP Port menu item to configure the TCP/IP port.

For each device on which the operation is started, a tab displays on the right panel (Test Lab area). The tab contains the operation information and status. Only one operation can be run at a time with ITDT-GE. The benefit in using tabs even for the single operation mode is that you get a history of operations as for each consecutive operation, a separate tab is opened.

## Graphical Edition - Scan menu commands



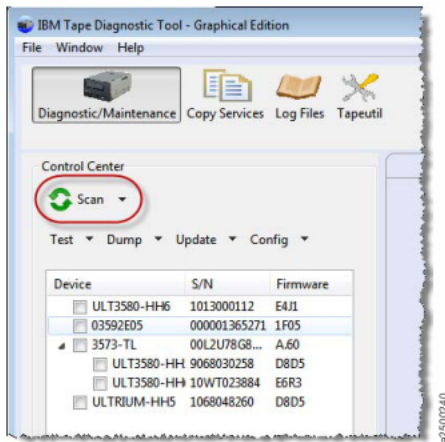
The following commands are described in this section:

- [Scan](#)
- [Health Test](#)
- [Configure TCP/IP Ports](#)
- [Dump](#)
- [Firmware Update](#)
- [Firmware Update - check for updates](#)
- [Encryption](#)
- [Full Write](#)
- [Tape Usage](#)
- [System Test](#)
- [Library Diagnostic Self-Test](#)
- [Library Media Screening](#)
- [LTFS Readiness Check](#)
- [Manual Inspection Record Entry](#)

---

## Scan

Figure 1. Scan



The scan function is used to discover all supported tape and library devices that are attached to the computer system. Then, they can be selected for the subsequent ITDT-GE operations. The scan function also serves as a connection test that can be used to verify correct attachment of the devices.

Note: When Scan is pressed for the first time, a dialog box is displayed that warns the user to stop all backup jobs. When the scan is finished, the device list is displayed in the Control Center area.

A scroll bar is available to show all the devices. When the device you want is displayed, select the device for test. Only one device can be selected.

---

## Specify Device Manually

The drop-down list next to Scan allows you to define a tape device and an associated changer device that must be used for testing. This function is a fast alternative to the Scan function where all Host Bus Adapters are discovered for supported tape and changer devices. The function supports the device name and HBTL addressing schema. For stand-alone tape devices the parameter Changer Device is not required.

Example device names:

**Windows:** \\.\tape0  
**Linux:** /dev/IBMtape0

Example HBTL (all platforms):

**H1-B0-T3-L0 or 1 0 3 0**

---

## Partial Scan

Partial Scan is used instead of Scan in cases when only a particular HBA or target ID range is used for device-discovering. The scan starts at "First device to scan" and checks for supported devices until "Last device to scan" is reached. Devices that are specified within "Exclude from scan" are skipped during the scan. If "First device to scan", "Last device to scan", or "Exclude from Scan" is not specified, ITDT performs a regular scan.

Partial Scan supports the device name and HBTL addressing schema.

Example device names:

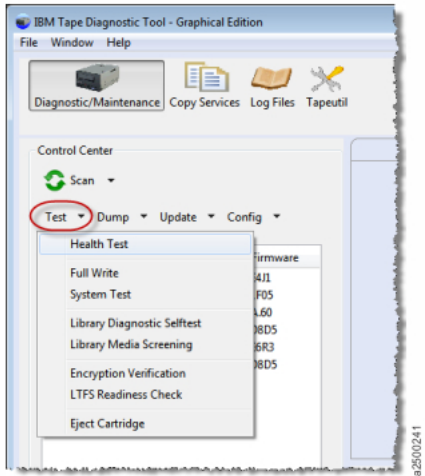
**Windows:** \\.\tape0  
**Linux:** /dev/IBMtape0

Example HBTL (all platforms):

**H1-B0-T3-L0 or 1 0 3 0**

# Health Test

Figure 1. Test



The Health Test function checks if the tape device is defective and outputs a pass/fail result.

Attention:

- 1. The test functionality erases user data on the cartridge that is used for the test.
- 2. The test can take from 15 minutes up to 2 hours.
- 3. The test runs only on tape drives, not on autoloaders or libraries.

To start the Health Test function, it is recommended that a new or rarely used cartridge is used. Scaled (capacity-reduced) cartridges must not be used to test the device.

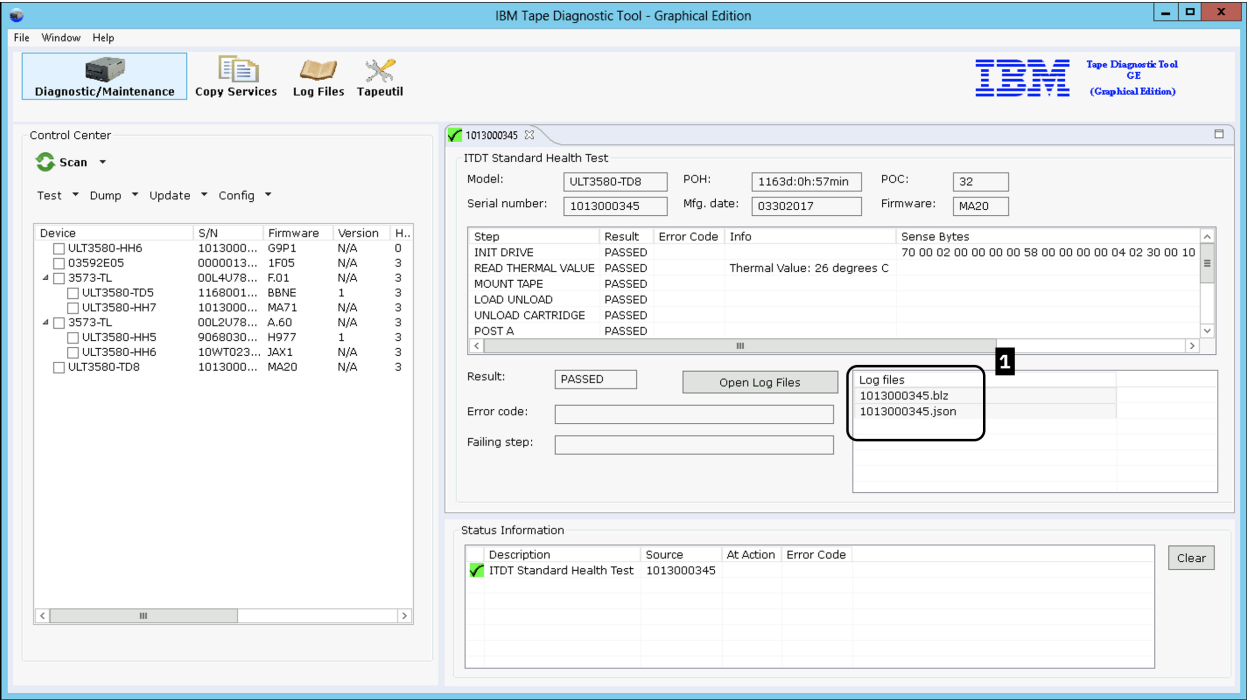
To test tape drives within a library, the library must be in online mode.

After the device is selected, start the Health Test function by selecting the Health Test menu item.

The Health Test function can be stopped by clicking Abort.

Note: It can take some time until the Health Test function stops.

Figure 2. Test results



Note: Information can be found in the .json/.blz files. See the Log files section ( 1 ).

The test sequence contains the following steps:

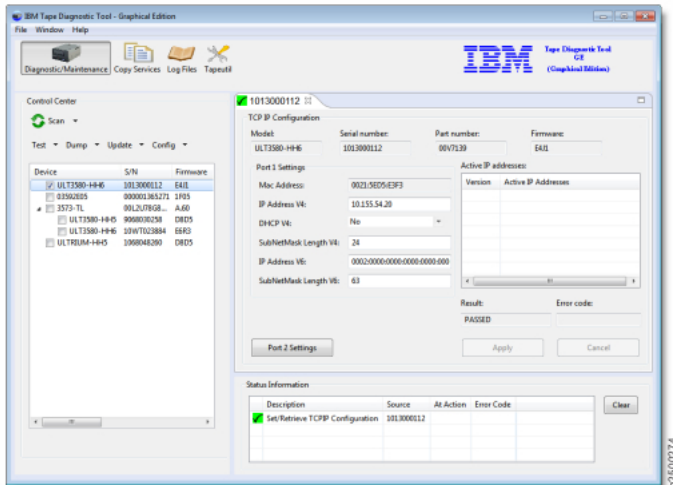
- 1. Initialize Drive
- 2. Read Thermal Sensor
- 3. Mount Medium
- 4. [Medium Qualification] – only if previous step indicated requirement

5. Load/Write/Unload/Read/Verify
6. POST A
7. Performance Test (run 2 times if first run failed with performance failure)
8. Unmount Medium
9. Read Thermal Sensor
10. Get FSC
11. Get Logs

## Configure TCP/IP Ports

For LTO 5, TS1140, and later drives, the ethernet port settings can be configured with the **Configure TCP/IP Port** command. The **Configure TCP/IP Port** command displays the current settings:

Figure 1. Configure TCP/IP Ports



After you click Apply, the new values are set and the updated values display.

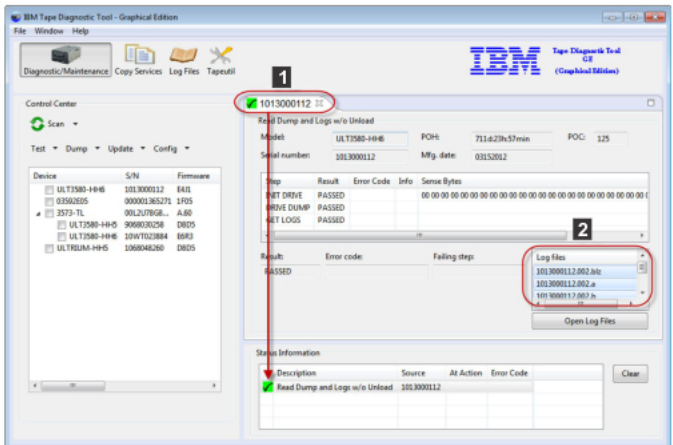
Note: Because earlier drive generations do not have an ethernet port, the **Configure TCP/IP Ports** command is rejected for these devices with the following message:  
TCP/IP configuration is not supported on this product.

## Dump

After the device you want to dump is selected, start the Dump function by selecting Dump, > Dump from the actions toolbar.

When the dump process is run on a tape library or autoloader other than the 3584/TS3500/TS4500, the Dump function stores 1 log file in the output folder of the program (\*.blz). For the 3584/TS3500/TS4500, a dump file (\*.a) is stored in the output folder. Both files start with the serial number of the device ( 1 ).

Figure 1. Dump results



Note: When the Dump function is run for tape libraries or autoloaders other than the 3584/TS3500/TS4500, the log file contains Log Sense and Mode Sense pages only. A Drive or 3584/TS3500/TS4500 dump contains more diagnostic information. ( 2 )

## Firmware Update

The Firmware Update upgrades the firmware of tape drives and tape libraries.

Note: See [con\\_a88u9\\_itdt\\_se\\_limits\\_all.html#con\\_a88u9\\_itdt\\_se\\_limits\\_all\\_3576](#) for information on how to update the firmware on a TS4500 tape library.

The following site is available for the latest firmware files: <http://www.ibm.com/storage/lto>

To do a Firmware Update, run the following steps:

1. Select the device that you want to update.
  2. Select the Update menu item.
  3. A standard file dialog opens to select the path of where the firmware update is located. The path is either the default input path or the location of the last successfully opened firmware file.
  4. Press OK on this file dialog to start the update on the selected device.
  5. During the firmware update, a firmware update progress screen is displayed, indicating the progress of the update.
- Attention: Once started, do not interrupt the firmware update.

The firmware update usually takes 3-5 minutes, but it can take up to 45 minutes for libraries.

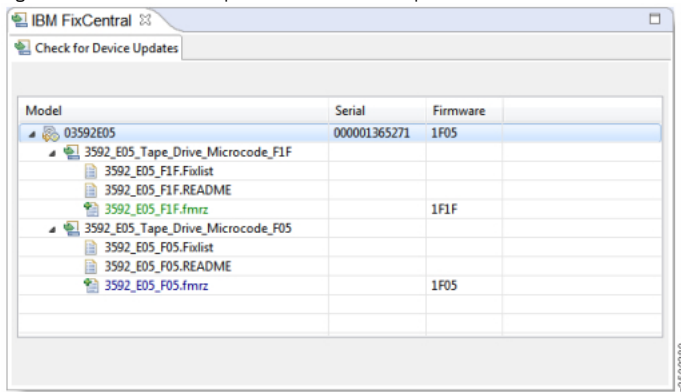
Note: If ITDT-GE detects a FIPS-certified drive firmware, it shows a warning dialog message. Before you continue, ensure that you use a FIPS-certified firmware to update the drive.

## Firmware Update - check for updates

ITDT-GE can Check for Device Updates for IBM® tape drives and IBM tape libraries. You can choose to select either one device or all devices by selecting the corresponding option from the drop down menu under Update in the ITDT Control Center.

The program connects to IBM FixCentral and identify updates for the devices. If the connection cannot be established or another problem occurred, the problem description is shown at the bottom of "FixCentral" Test Tab.

Figure 1. Check for Device Updates - FixCentral components



In the sample above, the drive has two FixCentral components. Each component has several items; such as code files and textual meta information.

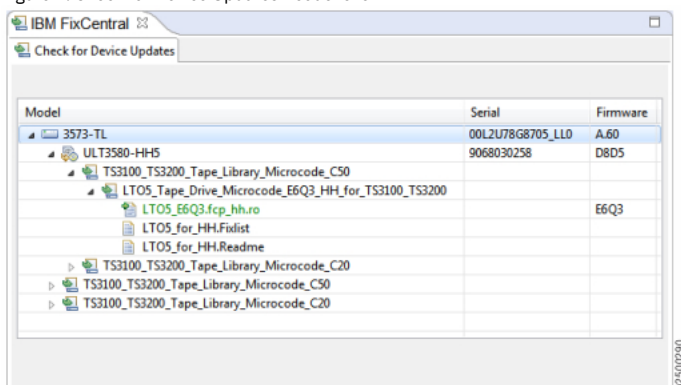
The code files (binary files) have a "+" in the icon to distinguish and can be downloaded either by double-clicking or by using the right mouse button. The text files (for example, readme files) can be viewed in the same way. A separate Test Tab is opened and the information shown.

For tape drives an available code is colored:

- GREEN: the code level is newer than the one of the drive.
- RED: the code level is older than the one of the drive.
- BLUE: the code level is the same as the one of the drive.

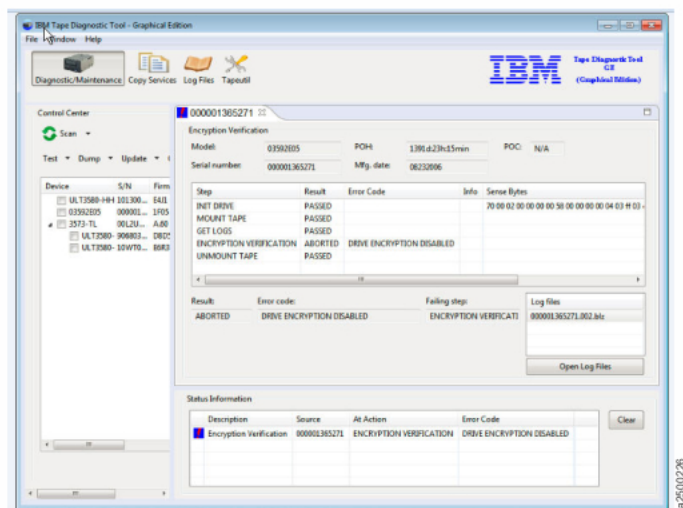
For automation drives (drives in a library), both devices are used. A code level for an automation drive is directly linked to a code level of a library.

Figure 2. Check for Device Updates - code level



## Encryption

Figure 1. Encryption



Note: The test that is shown on the graphic was run on a non-encrypted device and is showing a failure. The Encryption function is used to verify whether data on the cartridge is written encrypted. It reads both decrypted and raw data from the cartridge into two separate files on disk. The user can then verify that the data differs to ensure that encryption worked.

The Encryption function is only supported on encryption enabled drives and requires an encryption infrastructure, including the Encryption Key Manager (EKM) to be properly set up.

1. After the device you want to test is selected, start the encryption function by selecting **Test > Encryption Verification** from the actions toolbar.
2. ITDT-GE then shows the Encryption Verification screen. On this screen, the system requires the entry of the number of the start record and the amount of data (in KB) to be read.
3. Enter the required values and click OK to start the encryption.  
The Encryption function can be stopped by clicking Abort.  
Note: It can take some time until the Encryption function stops.

[Table 1](#) defines the abort codes.

## Full Write

The Full Write function writes the entire cartridge with a specified block size either with compressible or incompressible data and output performance data.

Attention:

1. The Full Write function erases data on the cartridge that is used for the test.
2. The Full Write function takes approximately 2 hours when incompressible data is written, less time for compressible data.
3. The Full Write function runs only on tape drives, not on autoloaders or libraries.

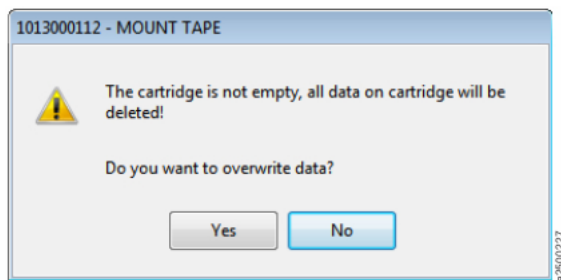
The Full Write test can be used to

- Demonstrate that the drive can write the full amount of data on a cartridge.
- Reveal system performance bottlenecks.  
Full Write runs the test with incompressible data first, then runs the test with compressible data. If no difference in overall performance is observed, a bandwidth limitation might be caused by the system or cabling or HBA.
- Identify system issues with compression.  
Compression is always turned on during the full write. When run with compressible data, the output shows the compression rate. If it is higher than 1.0 but your system is not able to compress data on the cartridge, check your device driver and software settings to see whether they disable compression.

To run a full write on tape drives within a library, the library must be in online mode.

1. After the device you want to write to is selected, start the Full Write function by selecting **Test > Full Write** from the actions toolbar.
2. Click OK to start the full write.
3. ITDT-GE then shows the Full Write screen. If no cartridge is inserted, ITDT-GE prompts you to insert a cartridge. Either insert a cartridge and click OK or stop by clicking Abort.  
Note: If ITDT-GE detects data on the cartridge, it shows the following message.

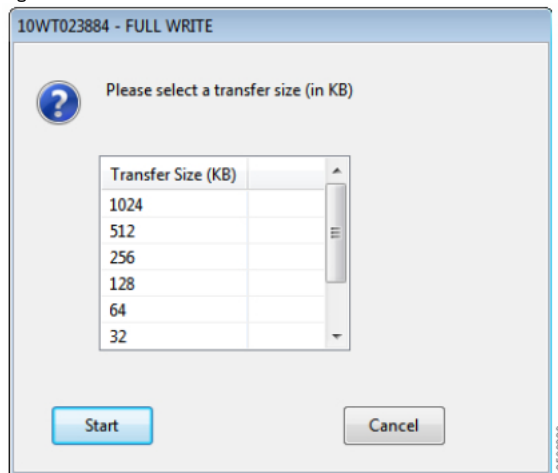
Figure 1. Overwrite data?



Click Yes to continue the test if you are sure that data on the cartridge can be overwritten. If you are unsure, click No to stop the test.

4. The system prompts for entry of a transfer size between 16 KB and the maximum block size that is supported by the system. Maximum value is 512 KB. Select the appropriate value for your system.

Figure 2. Transfer size



5. Select the type of data to write, either [C] Compressible or [I] Incompressible.

The full write can be stopped by clicking Abort.

Note: It can take some time until the full write actually stops.

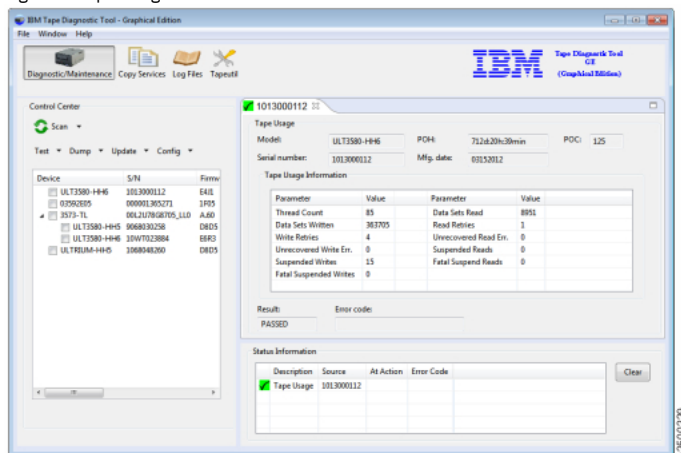
"Compressible = Yes" means that the data written was just zeros so that the data can be compressed by the drive with a maximum compression ratio.

"Compressible = No" means that a data pattern was written that the drive almost cannot compress at all. If the compression ratio is 1, the drive was not able to compress the data (equivalent to 1:1 compression ratio). If the compression ratio is 94.0, the drive was able to do 94:1 compression. It means that 94 bytes in the original data was compressed to 1 byte on the medium. 100.0 means 100 bytes is compressed down to 1 byte on the medium.

If all write operations are finished, ITDT-GE shows the performance statistics for the selected block size that is written on the cartridge, in the Status Information area. If an error occurred during the full write, data is only partially written.

## Tape Usage

Figure 1. Tape Usage



The Tape Usage function retrieves statistical data and error counters from a cartridge.

1. After the device you want to test is selected, start the Tape Usage function by selecting Dump > Tape Usage Log from the actions toolbar.
2. ITDT-GE then shows the tape usage screen. If no cartridge is inserted, ITDT-GE prompts you to insert a cartridge. Either insert a cartridge and click OK or stop by clicking Abort.

## System Test

The System Test is a short test that runs the following tests:

- Reveals system performance bottlenecks. Compressible data throughput values can reveal bandwidth limitations that are caused by the system or cabling or HBA.
- Measures performance variations across the different block sizes to find the ideal block size for the system configuration.

The System Test runs only on tape drives, not on autoloaders or libraries. To run a System Test on tape drives within a library, the library must be in online mode.

After the device you want to test is selected, start the System Test function by selecting Test > System Test from the actions toolbar.

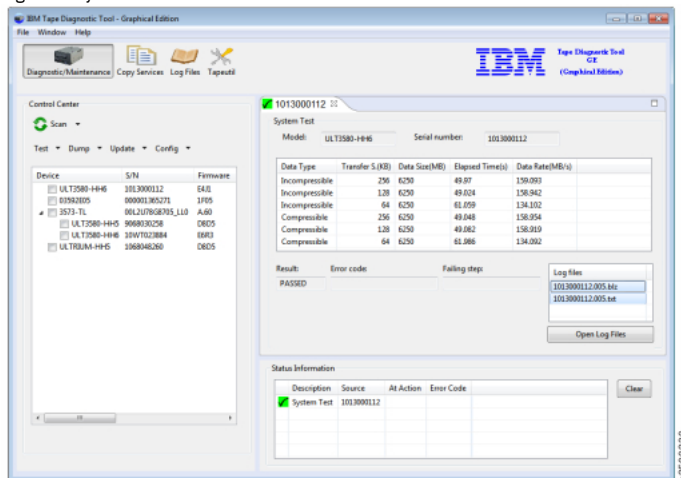
ITDT-GE then shows the System Test screen. If no cartridge is inserted, ITDT-GE prompts you to insert a cartridge. Either insert a cartridge and click OK or stop by clicking Abort.

Note: If ITDT-GE detects data on the cartridge, it shows the following message -

Cartridge not empty!  
Overwrite data?

Click Yes to continue the test if you are sure that data on the cartridge can be overwritten. If you are unsure, click No to stop the test.

Figure 1. System Test



The System Test is run as follows:

1. System Test determines the amount of data to write for each supported blocksize (a percentage of the cartridge is written for each blocksize).
2. The test determines the maximum supported blocksize of the system.
3. System Test writes the amount of data with all supported block sizes in powers of two down to 64 KB (values of 16 KB and 32 KB are not tested in cases where the capability of a system supports higher block sizes). It begins with the maximum supported blocksize that is detected before, first with incompressible, next with compressible data. Then, it shows performance data and a progress screen.
4. At the end of the test, a summary information is shown.

## Library Diagnostic Self-Test

The Library Self-Test starts and monitors the library-internal self-test. This test runs only on libraries and autoloaders, not on tape drives.

After the device you want to test is selected, start the Library Self-Test function by selecting Test > Library Diagnostic Self-Test from the actions toolbar.

At the end of the test, the results are shown in the Status Information area.

## Library Media Screening

The Automated Library Media Screening test generates dumps for each drive and cartridge within a library. It runs only on libraries (except TS3500/TS4500) and autoloaders, not on tape drives.

First, the test tries to read dump files from each drive that is installed from the library. Then, you can select one drive for loading the cartridges.

All cartridges of the I/O and storage slots are moved - one after the other from their source to the selected drive. A dump is taken and moved back to the source address. In the result screen, the dumps taken and the count of dumps are shown.

## LTFS Readiness Check

---

## Manual Inspection Record Entry

A manual inspection record can be generated if the device does not show in the device list. This test is intended for devices that are not recognized or have a technical problem that cannot be determined by ITDT-GE.

If a tape drive cannot be identified by using a device scan, the user can manually create a test record for the drive. The system prompts the user to run the SCSI/FC Wrap test for the drive (see the service manual for the drive). The results of the wrap test can be entered along with extra inspection information. The results are saved into binary and text output files that have the same format as the output files generated by the test.

1. From the Main Program menu, select File>>Manual Record.
2. Enter the required information to complete the items in the screen.
  - a. Enter the device serial number.
  - b. Enter the content of the Message Display.
  - c. Optionally, enter any information text.
3. After all information is entered, click OK.

The information is stored in a binary file (which can be used for further analysis), and in a human-readable text file. Both files are stored in the ITDT-GE output folder.

---

## Graphical Edition - Copy Services

With Copy and Migration Services, tape content can either be copied or moved

- From a cartridge to another cartridge
- From a cartridge to an image file, or
- From an image file to a cartridge.

On UNIX operating systems, a warning to check the file size limit is shown before an image file is created, if the maximum file size is below 2 GB.

The tool offers different use cases like copying data or migrating data from one generation to another. That is, data can be copied from a Gen4 cartridge to a Gen5 cartridge. Even migration from an LTO to IBM® Enterprise Tape Systems is supported. For an LTFS environment, the data can be copied or moved with the LTFS Physical copy that adjusts the LTFS parameter on the target to be unique. Such a copy can be used in the same LTFS environment as the source. To ensure maximum performance for tape to tape copy actions, ITDT allocates a read buffer that calculates to 2500 multiplied by the maximum supported system blocksize. If the system does not provide this buffer, the operation is aborted with `MEMORY ALLOCATION FAILED`. An infinite progress bar indicates the ongoing copy procedure. A dialog box confirms the completion.

---

## Tape Physical Copy

Creates a physical copy of a tape cartridge either to another cartridge or to an image file. The created cartridge has the same physical layout and contents as the source cartridge. Tape Physical Copy supports LTO and 3592 tape drives and can therefore be used for data migration. A copy from a previous generation to a newer generation cartridge and from an LTO to a 3592 works. However, the amount of used data from the source device must be equal or less than the capacity of the destination device. If the source cartridge is partitioned, the target must support the same partitioning sizes and amount. The amount of transferred data and the current data transfer rate is displayed every 3 - 5 minutes.

---

## LTFS Physical Copy

Creates a physical copy of an LTFS formatted cartridge. The LTFS specified parameters volumeuuid and VCI are adjusted during this copy operation. The created cartridge has the same physical layout as the origin cartridge. Expect the volumeuuid to be identical to the contents of the two cartridges. The amount of transferred data and the current data transfer rate is displayed every 3 - 5 minutes. When 'LTFS Physical Copy' is used with a non-LTFS formatted cartridge, the behavior is identical to Tape Physical Copy. For LTFS formatted cartridges, the volume identifier can be changed for a physical target by specifying New LTFS Volume Label.

---

## Verify

Verifies the physical contents of two cartridges. The physical data layout and the binary data are compared.

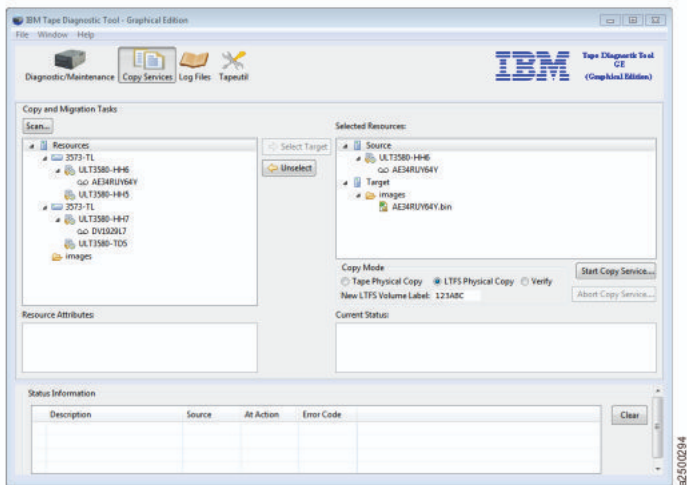
---

## Usage

By opening the "Copy Services" perspective, the user can run a 'Scan...' to discover attached devices.

Figure 1. Copy Services





A cartridge can be loaded or unloaded by double-clicking or pressing the right mouse button on the drive that is used for the copy tasks. When a cartridge is loaded, the cartridge Serial Number or the VOLSER in a library environment is shown. If a cartridge or an image file is selected on the left, it can be used as "Source" by pushing Select Source. After the "Source" is selected, the target can be chosen. To select the target, either a different cartridge in a different drive or the image file folder can be selected. By pressing Select Target, the target is set on the right.

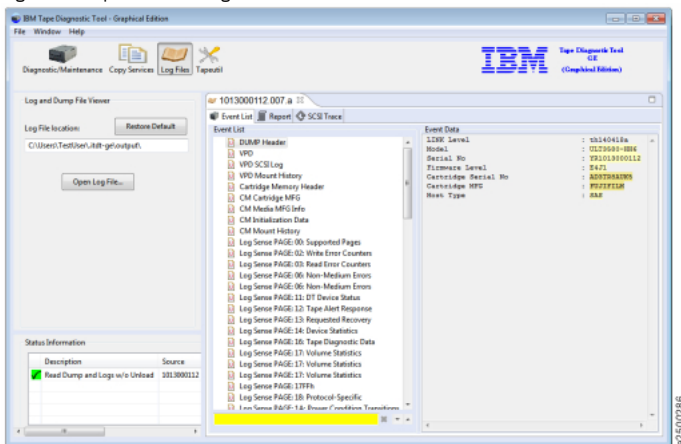
If the source and the target are selected, the Copy Services or a verification can be started regarding the used Mode: [Tape Physical Copy](#), [LTFS Physical Copy](#), or [Verify](#).

## Graphical Edition - visualizing log files

ITDT-GE offers the opportunity to visualize the content of DUMP and ITDT log files (BLOB files).

- Dump files can be retrieved with ITDT or any other tool that supports this function.
- Dump files that are generated by ITDT have the suffix ".a" or ".b".
- BLOB files are generated during the run of an ITDT test sequence, such as "Standard Test".
- BLOB files have the suffix ".blz".
- Both file types can be opened and visualized with ITDT-GE.

Figure 1. Graphic Edition: log view



## Opening a Dump or BLOB file

A Dump or a BLOB file can be opened either with the menu command **File/Open Log File** or by switching to the Log Files perspective and pressing Open Log File .... In both cases, a File Selection dialog opens where files can be selected for display. After the successful opening of a file, the data is shown.

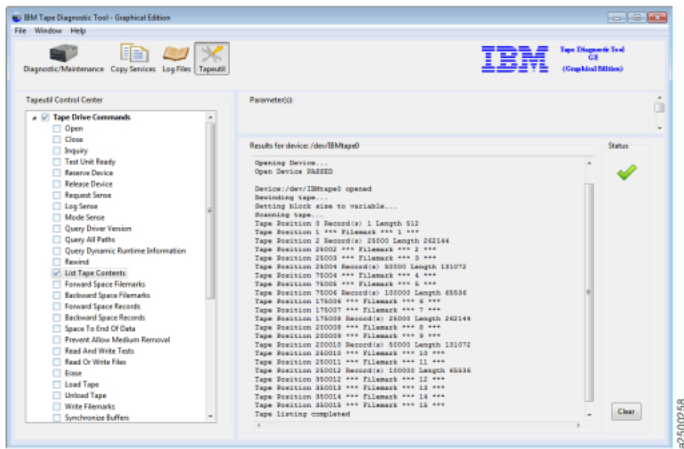
Two views are available for presenting the data - each at a separate panel.

- The Event List shows the events of this file on the left side. An event is a group of information and consists of 1 to n Elements shown on the right. By selecting one event list entry on the left, the corresponding data (event elements) is shown on the right.
- The Report panel offers the opportunity to generate a sublist of the available 'Event List' entries.

## Graphical Edition - Tapeutil menu commands

After the initial startup, ITDT-GE shows three figures under the top menu. After the Tapeutil option is selected, the following page opens.

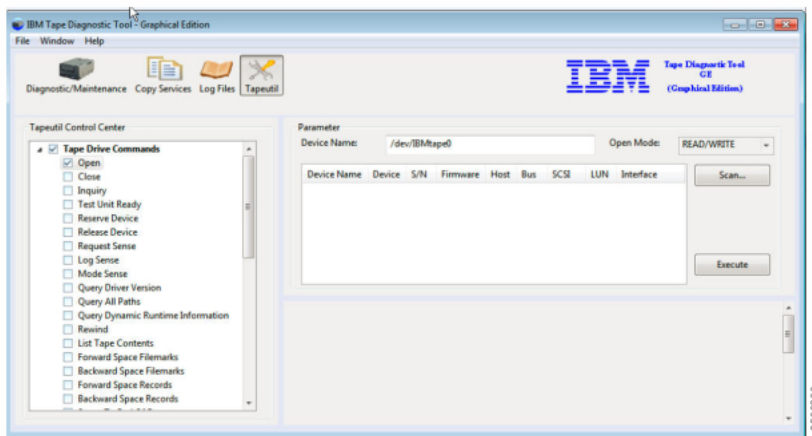
Figure 1. Tapeutil Control Center



On the left, the Tapeutil Control Center tree contains all Tapeutil commands for tape drives and tape libraries.

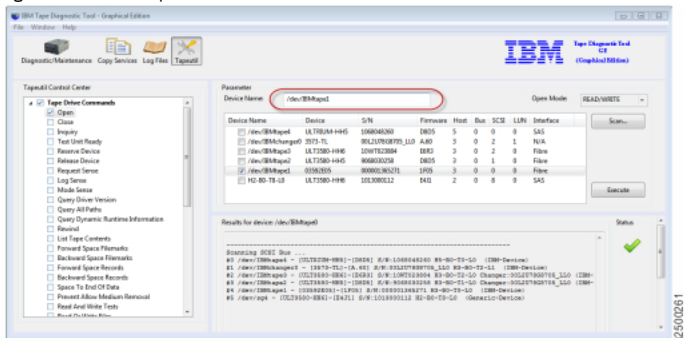
The commands from the General Commands and Service Aid Commands categories are duplicated into the two sections (to make the GUI navigation easier). After one category is expanded, the related commands open that allows users to select the command.

Figure 2. Command parameters



When the user presses Execute, the Results output is placed below the Parameter section:

Figure 3. Command parameter results



The **Open** command has a Scan ... button in the Parameter view. Pressing Scan... runs a scan on the host and shows the attached devices in the Result View at the bottom. This information is helpful to identify the right Device Name for the open function field Device Name.

Figure 4. Generic tapeutil scan

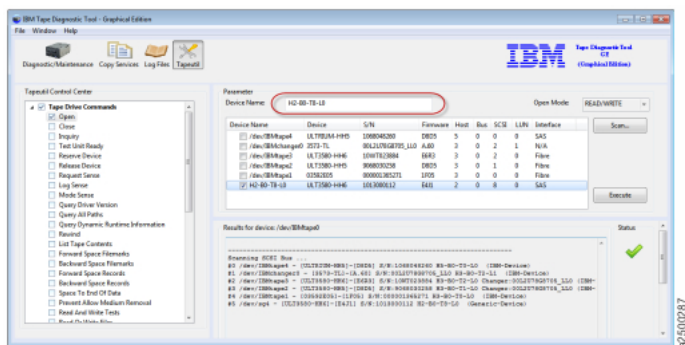
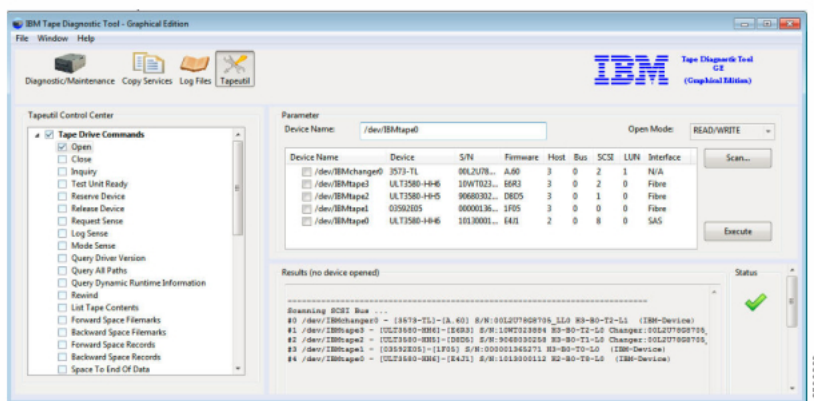


Figure 5. DD tapeutil scan



This screen layout stays within the Tapeutil perspective until the program is closed. Outputs of subsequent operations are added to the Results field. Commands that fail are indicated with a red cross in the Status area. Commands that succeed are indicated with a green check mark. The status area can be cleared by pressing Clear.

General commands

- [Open](#)
- [Close](#)
- [Inquiry](#)
- [Test Unit Ready](#)
- [Reserve Device](#)
- [Release Device](#)
- [Request Sense](#)
- [Log Sense](#)
- [Mode Sense](#)
- [Query Driver Version](#)
- [Display All Paths](#)

Tape drive specific commands

- [Rewind](#)
- [Forward Space Filemarks](#)
- [Backward Space Filemarks](#)
- [Servo Channel Calibration](#)
- [Forward Space Records](#)
- [Backward Space Records](#)
- [Space to End of Data](#)
- [Read and Write Tests](#)
- [Read or Write Files](#)
- [Erase](#)
- [Load Tape](#)
- [Unload Tape](#)
- [Write Filemarks](#)
- [Synchronize Buffers](#)
- [Query/Set Parameter](#)
- [Query/Set Position](#)
- [Query Encryption Status](#)
- [Display Message](#)
- [Display All Paths](#)
- [Report Density Support](#)
- [Test Encryption Path](#)

Tape library-specific commands

- [Element Information](#)
- [Position to Element](#)
- [Element Inventory](#)
- [Exchange Medium](#)

- [Move Medium](#)
- [Initialize Element Status](#)
- [Prevent/Allow Medium Removal](#)
- [Initialize Element Status Range](#)
- [Read Device IDs](#)
- [Read Cartridge Location](#)

Service aid commands

- [Configure TCP/IP Ports](#)
- [Dump/Force Dump/Dump](#)
- [Firmware Update](#)

Note: When a command is issued in Tapeutil mode for ITDT GE, Execute must be pressed before the action takes place.

- [Open](#)
- [Close](#)
- [Inquiry](#)
- [Test Unit Ready](#)
- [Reserve Device](#)
- [Release Device](#)
- [Request Sense](#)
- [Log Sense](#)
- [Mode Sense](#)
- [Query Driver Version](#)
- [Display All Paths](#)
- [Rewind](#)
- [Forward Space Filemarks](#)
- [Backward Space Filemarks](#)
- [Servo Channel Calibration](#)
- [Forward Space Records](#)
- [Backward Space Records](#)
- [Space to End of Data](#)
- [Read and Write Tests](#)
- [Read or Write Files](#)
- [Erase](#)
- [Load Tape](#)
- [Unload Tape](#)
- [Write Filemarks](#)
- [Synchronize Buffers](#)
- [Query/Set Parameter](#)
- [Query/Set Position](#)
- [Query Encryption Status](#)
- [Display Message](#)
- [Report Density Support](#)
- [Test Encryption Path](#)
- [Element Information](#)
- [Position to Element](#)
- [Element Inventory](#)
- [Exchange Medium](#)
- [Move Medium](#)
- [Initialize Element Status](#)
- [Prevent/Allow Medium Removal](#)
- [Initialize Element Status Range](#)
- [Read Device IDs](#)
- [Read Cartridge Location](#)
- [Configure TCP/IP Ports](#)
- [Dump/Force Dump/Dump](#)
- [Firmware Update](#)

---

## Open

When you select the **Open** command:

1. ITDT checks if a device is already opened.
2. Under Device Name:, enter the name of the device in the box.
3. In the Open Mode menu, select how to open the device (rw, ro, wo, append).
4. ITDT opens the device that you selected.

Note: Always use the Read Only mode when you are working with write-protected media.

---

## Close

When you select the **Close** command

1. ITDT checks if the device is already closed.

2. ITDT closes the device.

---

## Inquiry

When you select the **Inquiry** command

1. You are prompted for page code.
2. ITDT then displays a decoded format of a hexadecimal dump and prints a hexadecimal dump of the inquiry data.

---

## Test Unit Ready

When you select the **Test Unit Ready** (TUR) command, ITDT issues the **Test Unit Ready** *ioctl* command.

---

## Reserve Device

When you select the **Reserve Device** command, ITDT issues a reserve command for the device.

---

## Release Device

When you select the **Release Device** command, ITDT issues a release command for the device.

---

## Request Sense

When you select the **Request Sense** command

1. ITDT issues a **Request Sense** command.
2. ITDT then displays a decoded format of hexadecimal dump sense data and prints hexadecimal dump sense data.

---

## Log Sense

When you select the **Log Sense** command

1. Enter the page number, in hexadecimal, in the Page-Code field.
2. ITDT issues a **Log Sense** command and outputs a hexadecimal dump of that page.

---

## Mode Sense

When you select the **Mode Sense** command

1. Enter the page number, in hexadecimal, in the Page-Code field.
2. ITDT issues a **Mode Sense** command and outputs a hexadecimal dump of that page.

---

## Query Driver Version

When you select the **Query Driver Version** command

1. ITDT issues the required command to get the driver version.
2. ITDT prints the driver version.

---

## Display All Paths

When you select the **Display All Paths** command

1. ITDT issues an *ioctl* command.
2. ITDT outputs decoded path information for all paths.

---

## Rewind

When you select the **Rewind** command, ITDT issues the *ioctl* rewind command for the device.

---

## Forward Space Filemarks

When you select the **Forward Space Filemarks** command

1. Enter the amount of filemarks to forward space, in the Filemark-Count box.
  2. ITDT issues (extrinsic) *ioctl* command.
  3. The tape is positioned on the first block of the next file.
- 

## Backward Space Filemarks

When you select the **Backward Space Filemarks** command

1. Enter the amount of filemarks to backward space, in the Filemark-Count box.
  2. ITDT issues (extrinsic) *ioctl* command.
  3. The tape is positioned on the last block of the previous file.
- 

## Servo Channel Calibration

When you select the **Servo Channel Calibration** command:

1. ITDT issues the self diagnostic command for drive servo channel calibration.
  2. The channel calibration returns PASSED or FAILED regarding the result of the command.
- 

## Forward Space Records

When you select the **Forward Space Records** command

1. Enter the amount of records to forward space, in the Record-Count box.
  2. ITDT issues (extrinsic) *ioctl* command.
- 

## Backward Space Records

When you select the **Backward Space Records** command

1. Enter the amount of records to backward space, in the Record-Count box.
  2. ITDT issues (extrinsic) *ioctl* command.
- 

## Space to End of Data

When you select the **Space to End of Data** (EOD) command, ITDT issues the (extrinsic) *ioctl* command.

---

## Read and Write Tests

When you select the **Read and Write Tests** command, ITDT runs the following functions (Read and Write Test, Read Only Test, and Write Only Test). Three parameter fields have default values already in them. Next, a Test menu that gives you the option of Read Data from Tape, Write Data to Tape, and Read/Write/Verify.

Note: The default is a block size of 10240 bytes, a count of 20 blocks, and a repetition of 1. If the block size is zero, variable mode is used. With a fixed block size, a data amount of (block size \* blocks) is transferred with a single operation. This operation might get rejected if the total amount exceeds the transfer size of the system. In *SCSI Generic mode*, the actual transfer size equals to blocksize. In *IBM Device Driver mode*, the transfer size is blocksize \* blocks per read/write. In either case, the blocksize must be equal to or less than the maximum Host Bus Adapter supported transfer size. The maximum supported transfer size for *SCSI Generic* is 1048576 Bytes and for *IBM Device Driver* is 500 MB.

The following steps are run, depending on which test is selected.

- The Read/Write steps:
  1. Issues a Read Position.
  2. Sets block size.

3. Generates special pattern.
  4. Puts block id in bytes 0-3 of each block.
  5. Prints current block number, number of bytes and blocks.
  6. Issues write command.
  7. Prints updated statistics.
  8. If number of bytes written is different from requested bytes to write, stop (go to Step 19).
  9. Writes two filemarks.
  10. Backward spaces two filemarks.
  11. Backward spaces number of records written.
  12. Prints amount of data to read.
  13. Issues read command.
  14. If read error occurred or number of bytes read is different from requested number of bytes to read, go to Step 19.
  15. Compares data that is read with data written, show miscompares and if miscompares exist, stop (go to Step 19).
  16. If compare is OK, print OK message.
  17. Forward space one file mark.
  18. Repeat Steps 10 - 24 until all blocks are written, or go to Step 4 until all blocks are written.
  19. Prints current block id and total number of bytes written.
- The Read Only steps:
    1. Issues a Read Position.
    2. Sets block size.
    3. Generates special pattern.
    4. Print amount of data to read.
    5. Issues read command.
    6. If read error occurred or number of bytes read is different from requested number of bytes to read, stop (go to Step 19).
    7. Compares data that is read with buffer data, show miscompares and if miscompares exist, stop (go to Step 19).
    8. If compare is OK, print OK message.
    9. Repeat Steps 10 - 15 until all blocks are written, or go to Step 4 until all blocks are written.
    10. Prints current block id and total number of bytes read.
    11. Backward spaces number of records written.
    12. Prints amount of data to read.
    13. Issues read command.
    14. If read error occurred or number of bytes read is different from requested number of bytes to read, go to Step 19.
    15. Compares data that is read with data written, show miscompares and if miscompares exist, stop (go to Step 19).
    16. If compare is OK, print OK message.
    17. Forward space one file mark.
    18. Repeat Steps 10 - 24 until all blocks are written, or go to Step 4 until all blocks are written.
    19. Prints current block id and total number of bytes written.
  - The Write Only steps:
    1. Issues a Read Position.
    2. Sets block size.
    3. Generates special pattern.
    4. Put block id in bytes 0-3 of each block.
    5. Prints current block number, number of bytes and blocks.
    6. Issues write command.
    7. Prints updated statistics.
    8. If number of bytes written is different from requested bytes to write, stop (go to Step 10).
    9. Repeat Steps 5 - 9 until all blocks are written, or to Step 4 until all blocks are written.
    10. Print current block ID and total number of bytes written.

---

## Read or Write Files

When **Read or Write Files** is selected, a box under File Name: is where the path and name of the file is entered. Under that is a box named Number of records to read (0 for entire file). The default amount in the box is 100. Next, a menu bar under Test: gives you the choice of Read File from Tape or Write File to Tape. Once the Test is selected, Browse appears next to the File Name box to allow browsing for the needed file. When you select the **Read or Write Files** command, ITDT runs the following functions:

- Read steps:
  1. Prompts if to read a file from tape
  2. You are prompted for destination file name
  3. You are prompted for number of records to read (If you press Enter, the entire file is read.)
  4. Prints the file name to be opened
  5. Opens the file (r/w with large file support, 666 umask)
  6. Issues **Query Parameters** *ioctl* command, if it fails, quit
  7. Sets blocksize to maximum, variable blocksize mode
  8. Calculates the number of blocks to read.
  9. Prints number of records to read.
  10. ITDT read from tape.
  11. Writes to file, stop if data count is not equal to data count requested.
  12. If more data to read, go to Step 10.
  13. Prints statistics.
- Write steps:
  1. Prompts if to write a file to tape.
  2. You are prompted for the source file name.
  3. Prints the file name to be opened.
  4. Opens the file (r/o with large file support).
  5. Issues **Query Parameters** *ioctl* command, if it fails, quits.
  6. Sets blocksize to maximum, variable blocksize mode

7. Prints write statement.
8. Reads from file.
9. Writes to tape, stop if data counts written is not equal to data count requested.
10. Prints statistics.

---

## Erase

When you select the **Erase** command, ITDT issues the (extrinsic) *ioctl* command.

---

## Load Tape

ITDT issues a SCSI **Load** command to load a tape.

---

## Unload Tape

When you select the **Unload Tape** command

1. ITDT issues the (extrinsic) *ioctl* command.
2. The tape rewinds and then unloads.

---

## Write Filemarks

When you select the **Write Filemarks** command

1. In the Filemark-Count box, enter the number of filemarks to write.
2. ITDT issues the (extrinsic) *ioctl* command.

---

## Synchronize Buffers

When you select the **Synchronize Buffers** command, ITDT issues the *ioctl* command.

---

## Query/Set Parameter

When you select the **Query/Set Parameter** command

1. ITDT shows the changeable parameters.  
Note: The list of changeable parameters are operating system specific. For a list of changeable parameters, refer to [Table 1](#).
2. Select from the list of parameters that can be changed by clicking the choice.
3. ITDT requests prompt for parameter value (if required).
4. ITDT requests safety prompt (if required).
5. ITDT issues the *ioctl* command.

---

## Query/Set Position

When you select the **Query/Set Position** command

1. ITDT prints the current position and requests the new position.  
Note: ITDT does not distinguish between logical and physical position. It shows the current position and queries for the one to set, then sets the new position.
2. Enter the block id for where the tape must go. This block id must be entered in decimal. When the tape is set, the block id is printed in decimal with hexadecimal in parentheses.
3. ITDT issues the **Set Position** *ioctl* and returns the pass or fail results.
4. ITDT prints decoded logical position details.
5. ITDT issues **Query Physical Position** *ioctl* command.
6. ITDT prints decoded physical position details.
7. You are prompted for position to set (logical or physical) or to stop.
8. You are prompted for the block id in decimal or hexadecimal.
9. ITDT prints a summary.
10. ITDT issues the **Set Position** *ioctl* command.
11. Repeat steps 2 - 5.



---

## Query Encryption Status

When you select the **Query Encryption Status** command

1. ITDT issues **Get Encryption State** *ioctl* command.
2. ITDT displays encryption settings (Drive EC, Encryption Method, Encryption state).

---

## Display Message

When you select the **Display Message** command

1. ITDT provides Parameter boxes in which you can enter 1 or 2 messages up to 8 characters.  
Note: **Display Message** works only on drives that have a display pane, the 3590 and 3592 drives.
2. In the Type: menu, select which message (0 or 1) you want shown and if you want it to flash. There is also an alternate (alt) selection that alternates between messages.
3. ITDT issues the *ioctl* command.
4. ITDT prints the displayed message.

---

## Display All Paths

When you select the **Display All Paths** command

1. ITDT issues an *ioctl* command.
2. ITDT outputs decoded path information for all paths.

---

## Report Density Support

When you select the **Report Density Support** command

1. ITDT prints report status text for all supported media.
2. ITDT issues **Report Density Support** *ioctl* command to retrieve all supported media.
3. ITDT prints all requested reports. Data is printed in a decoded way. Scroll the screen to print each one of the following status texts:
  - Density name
  - Assigning organization
  - Description
  - Primary density code
  - Secondary density code
  - Write OK
  - Duplicate
  - Default
  - Bits per MM
  - Media Width
  - Tracks
  - Capacity (megabytes).
4. ITDT prints report status text for current media.
5. ITDT issues **Report Density Support** *ioctl* command to retrieve current media.
6. ITDT prints report data in a decoded way.

---

## Test Encryption Path

When you select the **Test Encryption Path** command

Note: Not supported for the HP-UX operating system.

1. ITDT prints status message that server configuration and connections are tested.
2. ITDT issues the **Encryption Diagnostics** *ioctl* command, Ping Diag.
3. ITDT prints number of servers available or error message.
4. ITDT issues the **Encryption Diagnostics** *ioctl* command, Basic Encryption Diag.
5. ITDT prints completion code or error message.
6. ITDT issues the **Encryption Diagnostics** *ioctl* command, Full Encryption Diag.
7. ITDT prints completion code or error message.

---

## Element Information

When you select the **Element Information** command:

1. ITDT issues the *ioctl* command.
2. ITDT shows
  - Number of robots
  - First robot address
  - Number of slots
  - First slot address
  - Number of I/E elements
  - First element address
  - Number of drives
  - First drive address

---

## Position to Element

When you select the **Position to Element** command:

1. In the Parameter boxes, the Transport element address must be entered, in decimal (picker).
2. Insert the Destination element address in decimal.
3. ITDT issues the *ioctl* command.

---

## Element Inventory

When you select the **Element Inventory** command:

1. ITDT issues the **Element Info** *ioctl* command.
2. ITDT issues the **Element Inventory** *ioctl* command.
3. ITDT displays decoded element inventory information.

---

## Exchange Medium

When you select the **Exchange Medium** command:

1. Insert source address into the Source address box in Decimal.
2. Insert the first destination address in decimal in the First destination address box.
3. Insert the second destination address in decimal in the Second destination address box.
4. ITDT issues the *ioctl* command.

---

## Move Medium

When you select the **Move Medium** command:

1. Insert source element address into the Source element address box in Decimal.
2. Insert the first destination element address in decimal in the First destination element address box.
3. Insert the second destination element address in decimal in the Second destination element address box.
4. ITDT issues the *ioctl* command.

---

## Initialize Element Status

When you select the **Initialize Element Status** command:

1. ITDT issues the *ioctl* command.
2. ITDT prints the command summary.

---

## Prevent/Allow Medium Removal

When you select the **Prevent/Allow Medium Removal** command:

1. Use the menu to either prevent or allow.
2. ITDT issues the *ioctl* command.

---

## Initialize Element Status Range

When you select the **Initialize Element Status Range** command:

1. Insert the first slot address in decimal in the provided box.
2. Insert the number of slots in the provided box.
3. ITDT issues the *ioctl* command.

---

## Read Device IDs

When you select the **Read Device IDs** command:

1. ITDT issues the **Element Info** *ioctl* command.
2. If no drive is present, ITDT prints `NO DRIVE PRESENT` and exits.
3. ITDT prints information for all drives.

---

## Read Cartridge Location

When you select the **Read Cartridge Location** command:

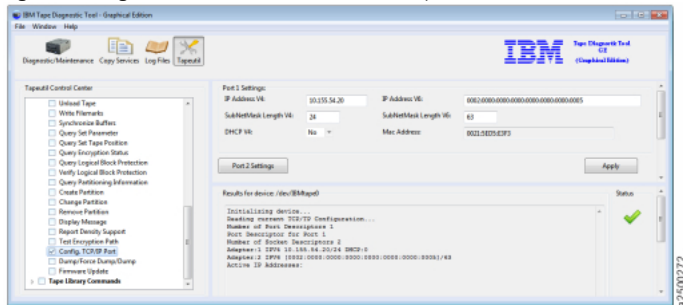
1. You are prompted for address of the first element.
2. If address is zero, print the error message and exit.
3. You are prompted for the number of elements.
4. If the number of elements is zero, print the error message and exit.
5. ITDT issues the **Element Info** *ioctl* command.
6. ITDT verifies that the address range is valid. Otherwise, print the error message and exit.
7. If no slots are found in Element Info data; print the error message and exit.
8. ITDT issues the **READ\_CARTRIDGE\_LOCATION** *ioctl* command.
9. ITDT prints decoded storage element information.

---

## Configure TCP/IP Ports

For LTO 5, TS1140, and later drives, the ethernet port settings can be configured with the **Configure TCP/IP Port** command. The **Configure TCP/IP Port** command displays the current settings:

Figure 1. Configure TCP/IP Ports command in the Graphical Edition



After you click Apply, the new values are set and the updated values display.

Note: Because earlier drive generations do not have an ethernet port, the **Configure TCP/IP Ports** command is rejected for these devices with the following message:

TCP/IP configuration is not supported on this product.

---

## Dump/Force Dump/Dump

When you select the **Dump/Force Dump/Dump** command:

1. ITDT retrieves the dump.
2. ITDT issues the **Force Dump** command.
3. ITDT retrieves the second dump.
4. ITDT displays the name of stored dump files and the output directory where they are stored. The dump filenames start with the serial number of the device.

---

## Firmware Update

When you select the **Firmware Update** command, browse to the microcode file to be used. ITDT runs the firmware update and displays progress status and result.

The following site is available for the latest firmware files: <http://www.ibm.com/support/fixcentral/>. Select System Storage > Tape Systems > Tape autoloaders and libraries or Tape drives.

---

## Verifying correct 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 verification, you must do the following procedure -

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 running commands according to the product documentation. See "IBM® Tape Product Publications" on page xiii.
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 in to the operating system as Administrator.
6. If device drivers are used by your device other than the ones documented in this manual, disable the other device drivers, and install or enable the drivers that are documented in this manual.
7. Start ITDT (for instructions see [Starting ITDT - Standard Edition](#)).
8. Scan for devices. Any devices that show up are properly attached.

---

## Platform-specific help

There is a problem determination section for each platform.

- AIX: [Problem determination](#)
- Linux: [Problem determination](#)
- Solaris: [Problem determination](#)
- Windows: [Problem determination](#)

---

## IBM technical support

If the problem persists after these procedures are followed, it is possible that an unexpected condition occurred in the driver's environment. In this case, contact your IBM service representative (1-800-IBM-SERV) and provide the following information to help IBM re-create and resolve the problem:

1. Machine type and model of your IBM tape product
2. Specific driver version
3. Description of the problem
4. System configuration
5. Operation that was running at the time the problem was encountered

---

## 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 ask what level of microcode you have on your hardware. If they believe that you need a new level of microcode, they might instruct you to install a newer level of microcode on your hardware. They can 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.

If your device is connected to a host system that has device or library support, you can also query the last 4 digits of the current level of microcode with software. Refer to [IBM Tape Diagnostic Tool \(ITDT\)](#). The unit must be powered On, configured properly, and ready. For information, refer to the appropriate chapter in this document (based on the operating system/platform) for details on how to have the device ready.

The following instructions are a guide to install another version of microcode on a 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 [Verifying correct attachment of your devices](#) to ensure that the drive is configured properly and ready.
2. Open ITDT and follow the instructions for downloading microcode. These instructions are in both the SE and the GE versions. In SE, it is available in all sections; scan menu under Firmware update, the tape utility (71) section, and the scripting (**ucode**) command.

---

## 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 may send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive

Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

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 cannot apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes are incorporated in new editions of the publication. IBM may make improvements and/or changes in the products and/or programs 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.

The ITDT-SE and ITDT-GE software uses Henry Spencer's regular expression library that is subject to the following copyright notice:

"Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice cannot be removed or altered.

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries (or regions), or both:

|              |               |                 |                       |
|--------------|---------------|-----------------|-----------------------|
| AIX®         | IBMLink       | RS/6000®        | System z®             |
| AIX 5L       | Magstar®      | S/390®          | Tivoli®               |
| FICON®       | Micro Channel | StorageSmart    | TotalStorage™         |
| HyperFactor® | Netfinity     | System i®       | Virtualization Engine |
| i5/OS        | POWER5        | System p        | xSeries               |
| iSeries      | ProtecTIER®   | System Storage® | z9                    |
| IBM          | pSeries       | System x        | zSeries               |

Adobe and Acrobat are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Itanium, and Pentium are trademarks of Intel Corporation in the United States, other countries (or regions), or both.

Java™ and all Java-based trademarks are trademarks of Oracle, Inc. in the United States, other countries, or both.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and Windows 2000 are 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).

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

## Terms and conditions for knowledge centers

Permissions for the use of these publications are granted subject to the following terms and conditions.

## IBM Privacy Policy

We intend to protect your personal information and to maintain its integrity. IBM® implements reasonable physical, administrative, and technical safeguards to help us protect your personal information from unauthorized access, use, and disclosure. For example, the products only provide IBM data about the asset usage and configuration and do not reflect private use of the asset. When diagnostics are required to be sent to IBM, and a problem is submitted, that data is routed directly to a secured infrastructure. Only individuals with a need to know are given access while working to resolve your problem. When appropriate, we also require that our suppliers protect such information from unauthorized access, use, and disclosure.

Visit the IBM Privacy Policy for additional information on this topic at <https://www.ibm.com/privacy/details/us/en/>.

## Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

## Personal use

---

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You cannot distribute, display, or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

## Commercial use

---

You can reproduce, distribute, and display these publications solely within your enterprise provided that all proprietary notices are preserved. You cannot make derivative works of these publications, or reproduce, distribute, or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## Rights

---

Except as expressly granted in this permission, no other permissions, licenses, or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions that are granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or as determined by IBM, the above instructions are not being properly followed.

You cannot download, export, or reexport this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM trademarks

---

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at <http://www.ibm.com/legal/copytrade.shtml>.

---

# IBM Tape Device Drivers Programming Reference

Welcome to the IBM Tape Device Drivers Programming Reference documentation, where you can find information about writing for applications to interfaces with IBM Tape Device Drivers on multiple platforms.

Last updated: December 16, 2019.

- [Introduction](#)  
These publications and URLs provide user information about writing for applications to interfaces for IBM® tape drives, medium changers, and library device drivers.
- [Common extended features](#)
- [AIX tape and medium changer device driver](#)
- [Linux tape and medium changer device driver](#)
- [Windows tape device drivers](#)
- [Notices](#)
- [How to send your comments](#)  
Your feedback is important in helping to provide the most accurate and highest quality information.
- [Terms and conditions for knowledge centers](#)  
Permissions for the use of these publications are granted subject to the following terms and conditions.

### Programming Reference

- [Introduction](#)
- [Common extended features](#)
- [AIX tape and medium changer device driver](#)
- [Linux tape and medium changer device driver](#)
- [Windows tape device drivers](#)

### Troubleshooting and support

- [IBM tape storage support](#)
- [IBM Support home page](#)

### More information

- [IBM Tape Device Drivers Programming Reference \(Legacy\)](#)
- [IBM TS4500 tape library documentation](#)
- [IBM TS2900 tape autoloader documentation](#)
- [IBM TS2270 tape drive documentation](#)
- [IBM TS2280 tape drive documentation](#)
- [IBM Community \(Community platform\)](#)
- [IBM Support content \(Product support\)](#)
- [Redbooks home page](#)

©Copyright IBM Corporation 2017-2021. Last updated: December 16, 2019.

---

## Introduction

These publications and URLs provide user information about writing for applications to interfaces for IBM® tape drives, medium changers, and library device drivers.

---

## Purpose

The IBM tape and medium changer device drivers are designed specifically to take advantage of the features that are provided by the IBM 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 unmount), and 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.

The most current information on hardware and software requirements for IBM tape and medium changer device drivers can be found in the individual platform readme files or with the subsequent links.

---

## Hardware requirements

The tape drivers and the IBM Tape Diagnostic Tool (ITDT) are developed to support various versions of different platforms. For the latest support, refer to the Interoperation Center website - <http://www.ibm.com/systems/support/storage/config/ssic/>.

Note: 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 either tape, disk, or both) where disk and tape subsystems share a common HBA, stability problems are observed. These issues are resolved by separating disk and tape I/O streams onto separate HBAs and by using SAN zoning to minimize contention. IBM is focused on assuring server/storage configuration interoperability. It is strongly recommended that your overall implementation plan includes provisions for separating disk and tape workloads.

For information about this issue, see the following Redbook, section 4.1.3 in <http://www.redbooks.ibm.com/abstracts/sg246502.html?Open>.

---

## Software requirements

If you use a third-party application, consult with your application provider about the compatibility with IBM tape device drivers.

For detailed driver requirements for each operating system, refer to the appropriate chapter.

---

## IBM tape products

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

- IBM TS2250/TS2260/TS2270/TS2280 tape drive
- IBM TS2350/TS2360 tape drive
- IBM TS1140/TS1150/TS1155/TS1160 tape drive (Enterprise)
- IBM TS3500 and TS4500 tape library (also known as IBM UltraScalable tape library 3584)
- IBM TS4300 tape library

---

## Related information

Reference material, including the Adobe pdf version of this publication, is available at <http://www-01.ibm.com/support/docview.wss?uid=ssg1S7003032>.

A companion publication that covers installation and user aspects for the device drivers is *IBM Tape Device Drivers: Installation and Users Guide*, GC27-2130-00, at <http://www-01.ibm.com/support/docview.wss?uid=ssg1S7002972>.

---

## AIX®

The following URL points to information about IBM System p (also known as pSeries) servers: <http://www-1.ibm.com/servers/eserver/pseries>.

---

## Linux

The following URLs relate to Linux® distributions: <http://www.redhat.com> and <http://www.suse.com>.

---

## Microsoft Windows

The following URL relates to Microsoft Windows systems: <http://www.microsoft.com>.

---

## Additional information

The following publication contains information that is related to the IBM tape drive, medium changer, and library device drivers: *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*.

---

## Common extended features

- [Tape drive functions and device driver ioctls](#)
- [Media partitioning](#)

- [Data safe \(append-only\) mode](#)
- [Read Position long/extended form and Locate\(16\) commands](#)
- [Logical Block Protection](#)
- [Programmable Early Warning \(PEW\)](#)
- [Log Sense page and subpage](#)
- [Mode Sense page and subpage](#)
- [Verify Tape](#)
- [RAO - Recommended Access Order](#)

## Tape drive functions and device driver ioctls

Beginning with the TS1140 (JAG 4), TS2250, and TS2350 (LTO 5) generation of tape drives, functions are supported that previous generations of LTO and JAG tape drives do not support. The device drivers provide *ioctls* that applications can use for these functions. Refer to the appropriate platform section for the specific *ioctls* and data structures that are not included in this section.

- Media Partitioning  
Supported tape drives: LTO 5 and JAG 4 and later models
- Data Safe (Append-Only) Mode  
Supported tape drives: LTO 5 and JAG 4 and later models
- Read Position SCSI Command for Long and Extended forms  
Supported tape drives: LTO 5 and JAG 4 and later models
- Locate(16) SCSI Command  
Supported tape drives: LTO 5 and JAG 4 and later models
- Logical Block Protection  
Supported tape drives: LTO 5 and JAG 2/3/4 and later models
- Programmable Early Warning (PEW)  
Supported tape drives: LTO 5 and JAG 2 and later models
- Log Sense Page and Subpage  
Supported tape drives: LTO 5 and JAG 3 and later models
- Mode Sense Page and Subpage  
Supported tape drives: LTO-4 and JAG 2 and later models
- Verify Tape  
Supported tape drives: LTO5 and JAG 2 and later models

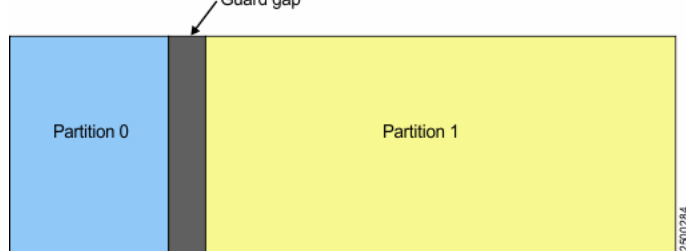
## Media partitioning

There are two types of partitioning: Wrap-wise partitioning and Longitudinal partitioning (maximum 2 partitions).

Figure 1. Wrap-wise partitioning



Figure 2. Longitudinal partitioning



In Wrap-wise partitioning, media can be partitioned into 1 or 2 partitions for LTO 5 and 1 - 4 partitions for TS1140. For later generations, see drive documentation for the number of partitions supported. The data partition (the default) for a single partition always exists as partition 0. WORM media cannot be partitioned.

The *ioctls* the device drivers provide for tape partitioning are

- Query Partition



The Query Partition ioctl returns the partition information for the current media in the tape drive. It also returns the current active partition the tape drive is using for the media.

Note: If the Create Partition ioctl fails, then the Query Partition ioctl does not return the correct partition information. To get the correct information, the application must unload and reload the tape again.

- Create Partition

The Create Partition ioctl is used to format the current media in the tape driver to either 1 or 2 partitions. When two partitions are created, the FDP, SDP, or IDP partition type is specified by the application. The tape must be positioned at the beginning of tape (partition 0 logical block id 0) before this ioctl is used or the ioctl fails.

If the number\_of\_partitions field to create in the ioctl structure is one partition, all other fields are ignored and not used. The tape drive formats the media by using its default partitioning type and size for a single partition.

When the type field in the ioctl structure is set to either FDP or SDP, the size\_unit and size fields in the ioctl structure are not used. When the type field in the ioctl structure is set to IDP, the size\_unit and size fields are used to specify the size for each partition. One of the two partition sizes for either partition 0 or 1 must be specified as 0xFFFF to use the remaining capacity. The other partition is created by using the size\_unit and size field for the partition.

- Set Active Partition

The Set Active Partition ioctl is used to position the tape drive to a specific partition. It becomes the current active partition for subsequent commands and a specific logical block id in the partition. To position to the beginning of the partition, the logical\_block\_id field in the ioctl structure must be set to 0.

---

## Data safe (append-only) mode

Data safe (append-only) mode sets the drive into a logical WORM mode so any non-WORM tape when loaded is handled similarly to a WORM tape. After data or filemarks are written to the tape, it cannot normally be overwritten. New data or filemarks can be appended only at the end of previously written data. Data safe mode applies only to drive operation. When a non-WORM tape is unloaded, it does not change and is still a non-WORM tape.

Conditions exist when the drive is in data safe mode an application might want to explicitly overwrite previously written data by issuing a **write**, **write filemark**, or **erase** command. These commands are referred to as write type commands. An application might also want to explicitly partition the tape with the Create Partition ioctl that issues a **format** command. The drive supports a new **Allow Data Overwrite SCSI** command for this purpose.

The ioctls that the device drivers provide for data safe mode are

- Querying and setting data safe mode

All platform device drivers except Windows added a data safe mode parameter to existing ioctls that are used to query or set tape drive parameters. The Windows device driver added two new ioctls to query or set data safe mode.

A query ioctl returns the current drive mode, either data safe mode off (normal mode) or data safe mode on. A set ioctl sets the drive to either data safe mode off (normal mode) or data safe mode on. Data safe mode can be set whether a tape is loaded in the drive or not. Data safe mode can be set back to normal mode only when a tape is not currently loaded in the drive.

- Allow Data Overwrite

The Allow Data Overwrite ioctl is used to allow previously written data on the tape to be overwritten when data safe mode is enabled on the drive, for a subsequent write type command, or to allow a format command with the Create Partition ioctl.

To allow a subsequent write type command, the tape position must be set to the correct partition and logical block address within the partition before the ioctl is used. The partition\_number and logical\_block\_id fields in the ioctl structure must be set to that partition and logical block ID. The allow\_format\_overwrite field in the ioctl structure must be set to 0.

To allow a subsequent Create Partition ioctl to format the tape, the allow\_format\_overwrite field in the ioctl structure must be set to 1. The partition\_number and logical\_block\_id fields are not used. But, the tape must be at the beginning of tape (partition 0 logical block id 0) before the Create Partition ioctl is issued.

---

## Read Position long/extended form and Locate(16) commands

Because of the increased tape media capacity and depending on the block sizes and number of files an application can write on tape, the 4-byte fields such as the logical block id the current **Read Position** command (referred to as the short form) that returns 20 bytes might overflow. The same applies to the **Locate(10)** command for the logical block id.

LTO 5 and later supports new forms of the existing **Read Position** command in addition to the current short form. The short form continues to return 4-byte fields in 20 bytes of return data. The long form returns 8-byte fields in 32 bytes of return data with the current position information for the logical block id and logical filemark. The extended form returns 8-byte fields in 32 bytes of return data with the current position information for the logical block id and buffer status. The format of return data in the **Read Position** command is specified by using a service action field in the Read Position SCSI CDB.

LTO 5 and later also supports the **Locate(16)** command that uses 8-byte fields. This command can either position the tape to a logical block id or a logical filemark by setting the dest\_type field in the Locate(16) SCSI CDB. After the locate command completes, the tape is positioned at the BOP side of the tape.

The ioctls the device drivers provide are

- Read Tape Position

The Read Tape Position ioctl returns the **Read Position** command data in either the short, long, or extended form. The form to be returned is specified by setting the data\_format field in the ioctl structure.

- Set Tape Position

The Set Tape Position ioctl issues a **Locate(16)** command to position the tape in the current active partition to either a logical block id or logical filemark. The logical\_id\_type field in the ioctl structure specifies either a logical block or logical filemark.

---

## Logical Block Protection

The ioctls the device drivers provide are

- Query Logical Block Protection

This ioctl queries whether the drive can support this feature, what lbp method is used, and where the protection information is included.

The lbp\_capable field indicates that the drive has the logical block protection (LBP) capability or not. The lbp\_method field is shown if LBP is enabled and what the protection method is. The LBP information length is shown in the lbp\_info\_length field. The fields of lbp\_w, lbp\_r, and rbdp present that the protection information is included in write, read, or recover buffer data. The rbdp field is not supported for the LTO drive.

- Set Logical Block Protection

This ioctl enables or disables Logical Block Protection, sets up what method is used, and where the protection information is included.

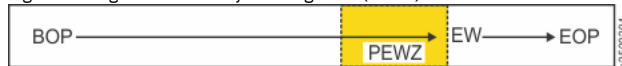
The lbp\_capable field is ignored in this ioctl by the tape driver. If the lbp\_method field is 0 (**LBP\_DISABLE**), all other fields are ignored and not used. When the lbp\_method field is set to a valid non-zero method, all other fields are used to specify the setup for LBP.

---

## Programmable Early Warning (PEW)

With the tape parameter, the application is allowed to request the tape drive to create a zone that is called the programmable early warning zone (PEWZ) in front of Early Warning (EW).

Figure 1. Programmable Early Warning Zone (PEWZ)



This parameter establishes the programmable early warning zone size. It is a 2-byte numerical value that specifies how many MB before the standard end-of-medium early warning zone to place the programmable early warning indicator. If the value is set to a positive integer, a user application is warned that the tape is running out of space when the tape head reaches the PEW location. If pew is set to 0, then there is no early warning zone and the user is notified only at the standard early warning location.

---

## Log Sense page and subpage

This ioctl of the SIOC\_LOG\_SENSE10\_PAGE issues a **Log Sense(10)** command and returns log sense data for a specific page and subpage. This ioctl command is enhanced to add a subpage variable from the log sense page. It returns a log sense page or subpage from the device. The wanted page is selected by specifying the page\_code or subpage\_code in the structure. Optionally, a specific parm pointer, also known as a parm code, and the number of parameter bytes can be specified with the command.

---

## Mode Sense page and subpage

This ioctl of the SIOC\_MODE\_SENSE issues a **Mode Sense(10)** or **(6)** command and returns the whole mode sense data. The data includes the header, block descriptor, and page code for a specific page or subpage from the device.

---

## Verify Tape

The ioctl of VERIFY\_DATA\_TAPE issues the **VERIFY** command to cause data to be read from the tape and passed through the drive's error detection and correction hardware. This action determines whether data can be recovered from the tape. Also, whether the protection information is present and validates correctly on logical block on the medium. The driver returns a failure or success signal if the **VERIFY SCSI** command is completed in a Good SCSI status. The **Verify** command is supported on all LTO libraries. Verify to EOD (ETD) or verify by filemark (VBF) is supported on drives that support Logical Block Protection (LBP).

---

## RAO - Recommended Access Order

The 3592 E07 implements a function that is called Recommended Access Order. This function provides the capability to improve multiple block recall and retrieval times. It provides an application with the optimized order in which a list of blocks must be recalled to minimize the required total time period.

An application uses the **GRAO** command to request that the drive generate a recommended access order for the User Data Segments that are sent in this command. After a **GRAO** command completes, use the **RRAO** command to receive the results.

---

## QUERY\_RAO\_INFO IOCTL

The IOCTL queries the maximum number and size of User Data Segments (UDS) that are supported from tape drive and driver for the wanted **uds\_type**: with or without geometry, the geometry can be used to build a representation of the physical layout of the UDS on tape. The application calls this IOCTL before the **GENERATE\_RAO** and **RECEIVE\_RAO** IOCTLs are issued. The return in this IOCTL is to be used by the application to limit the number of UDS requested in calls to the **GENERATE\_RAO** IOCTL.

## GENERATE\_RAO

---

The IOCTL is called to send a GRAO list (UDS's descriptors list) to request the drive to generate a Recommending Access Order list. The process method to create a RAO list is either 1 or 2. 1 does not reorder the UDS's list, but does calculate the estimated locate time for each UDS in the list, and 2 reorders the UDS's list and calculates the estimated locate time for each UDS in its resultant position. The type of UDS is either with or without the geometry. The `uds_number` must be not larger than `max_host_uds_number` returned in the **QUERY\_RAO\_INFO** IOCTL. A UDS descriptor has a name, partition number and beginning and ending logical object identifiers.

## RECEIVE\_RAO

---

After a **GENERATE\_RAO** IOCTL is completed, the application calls the **RECEIVE\_RAO** IOCTL to receive a recommended access order of UDS from the drive. The application must allocate to the buffer an accurate size to receive the list.

The `grao` list, for the `generate_rao` and `receive_rao` structures, is in the following format when lists are sent or received. The structures for the headers and UDS segments are not provided by the device driver but is the responsibility of the calling application. It is defined in the *IBM Enterprise Tape System 3592 SCSI Reference*.

```
-- List Header
-- UDS Segment Descriptor (first)
.....
-- UDS Segment Descriptor (last)
```

---

## AIX tape and medium changer device driver

This chapter provides an introduction to the IBM® AIX® Enhanced Tape and Medium Changer Device Driver (Atape) programming interface to IBM TotalStorage™ (formerly Magstar®) and System Storage® tape and medium changer devices.

- [Software interface for tape devices](#)
- [Software interface for medium changer devices](#)
- [Special files](#)
- [Device and volume information logging](#)
- [Persistent reservation support and IOCTL operations](#)
- [General IOCTL operations](#)
- [Tape IOCTL operations](#)
- [Medium changer IOCTL operations](#)
- [Return codes](#)

---

## Software interface for tape devices

The AIX® tape and medium changer device driver provides the following entry points for tape devices.

|       |                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Open  | This entry point is driven by <code>open</code> , <code>openx</code> , and <code>creat</code> subroutines.                                                                                                                                                                                                                                                                                                   |
| Write | This entry point is driven by <code>write</code> , <code>writex</code> , <code>writex</code> , and <code>writex</code> subroutines.                                                                                                                                                                                                                                                                          |
| Read  | This entry point is driven by <code>read</code> , <code>readv</code> , <code>readx</code> , and <code>readx</code> subroutines.                                                                                                                                                                                                                                                                              |
| Close | This entry point is driven explicitly by the <code>close</code> subroutine and implicitly by the operating system at program termination.                                                                                                                                                                                                                                                                    |
| ioctl | This entry point provides a set of tape and SCSI-specific functions. It allows AIX applications to access and control the features and attributes of the tape device programmatically. For the medium changer devices, it also provides a set of medium changer functions that is accessed through the tape device special files or independently through an extra special file for the medium changer only. |
| Dump  | This entry point allows the use of the AIX dump facility with the driver.                                                                                                                                                                                                                                                                                                                                    |

The standard set of AIX device management commands is available. The **chdev**, **rmdev**, **mkdev**, and **lsdev** commands are used to bring the device online or change the attributes that determine the status of the tape device.

---

## Software interface for medium changer devices

The AIX® tape and medium changer device driver provides the following AIX entry points for the medium changer devices.

|       |                                                                                                                                                                                                                 |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Open  | This entry point is driven by <code>open</code> and <code>openx</code> subroutines.                                                                                                                             |
| Close | This entry point is driven explicitly by the <code>close</code> subroutine and implicitly by the operating system at program termination.                                                                       |
| IOCTL | This entry point provides a set of medium changer and SCSI-specific functions. It allows AIX applications to access and control the features and attributes of the tape system robotic device programmatically. |

The standard set of AIX device management commands is available. The **chdev**, **rmdev**, **mkdev**, and **lsdev** commands are used to bring the device online or change the attributes that determine the status of the tape system robotic device.

## 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 (with other files unique to the Atape driver), are in the /dev directory.

- [Special files for tape devices](#)
- [Special files for medium changer devices](#)
- [Opening the special file for I/O](#)
- [The extended open operation](#)
- [Writing to the special file](#)
- [Reading from the special file](#)
- [Reading with the TAPE\\_SHORT\\_READ extended parameter](#)
- [Reading with the TAPE\\_READ\\_REVERSE extended parameter](#)
- [Closing the special file](#)

## 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. In addition to the tape special files, a special file is provided to tape devices that allow access to the medium changer as a separate device. See [Table 1](#). The asterisk (\*) represents a number that is assigned to a particular device (such as rmt0).

Table 1. Special files for tape devices

| Special File Name           | Rewind on Close <sup>1</sup> | Retension on Open <sup>2</sup> | Bytes per Inch <sup>3</sup> | 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 <sup>4</sup>   | 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 <sup>5</sup> | Yes                          | No                             | N/A                         | No            | No              |
| /dev/rmt*.smc <sup>6</sup>  | N/A                          | N/A                            | N/A                         | N/A           | N/A             |

Note:

1. The Rewind on Close special files for the Ultrium tape drives write filemarks under certain conditions before rewinding. See [Opening the special file for I/O](#).
2. The Retension on Open special files rewind the tape on open only. Retensioning is not completed because these tape products run 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\*.10 file bypasses normal close processing, and the tape is left at the current position.
5. 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 that is attached to it, and the device driver returns a successful completion. Read and write system calls return the requested number of bytes. This file can be used for application development or debugging problems.
6. The rmt\*.smc file can be opened independently of the other tape special files.

For tape drives with attached SCSI 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 SCSI device.

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 that are designed for a tape device

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

## 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 1](#) shows the attributes of the special file. The asterisk (\*) represents a number that is 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.

Table 1. Special files

| Special file name | Description                                 |
|-------------------|---------------------------------------------|
| /dev/smc*         | Access to the medium changer robotic device |
| /dev/smc*.null    | Pseudo medium changer device                |

Note: The smc\*.null file is a pseudo device similar to the /dev/null AIX® special file. The commands can be issued to this file without a real device that is attached to it, and the device driver returns a successful completion. This file can be used for application development or debugging problems.

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

- Read
- Write
- Commands that are designed for a tape device

## Opening the special file for I/O

Several options are available when a file is opened for access. These options, which are known as O\_FLAGS, affect the characteristics of the opened tape device or the result of the open operation. The **Open** command is

```
tapefd=open("/dev/rmt0",O_FLAGS);
smcfd=open("/dev/smc0",O_FLAGS);
```

The O\_FLAGS parameter has the following flags.

- O\_RDONLY  
This flag allows only operations that do not change the content of the tape. The flag is ignored if it is used to open the smc special files.
- O\_RDWR  
This flag allows complete access to the tape. The flag is ignored if it is used to open the smc special files.
- O\_WRONLY  
This flag does not allow the tape to be read. All other operations are allowed. The flag is ignored if it is used to open the smc special files.
- O\_NDELAY or O\_NONBLOCK  
These two flags complete the same function. The driver does not wait until the device is ready before it opens and allows commands to be sent. If the device is not ready, subsequent commands (which require that the device is ready or a physical tape is loaded) fail with ENOTREADY. Other commands, such as gathering the inquiry data, complete successfully.
- O\_APPEND  
When the tape drive is opened with this flag, the driver rewinds the tape. Then, it seeks to the first two consecutive filemarks, and places the initial tape position between them. This status is the same if the tape was previously opened with a No Rewind on Close special file. This process can take several minutes for a full tape. The flag is ignored if it is used to open the smc special files.

This flag must be used with the O\_WRONLY flag to append data to the end of the current data on the tape. The O\_RDONLY or O\_RDWR flag is illegal in combination with the O\_APPEND flag.

Note: This flag cannot be used with the Retension on Open special files, such as rmx.2.

If the open system call fails, the *errno* value contains the error code. See [Return codes](#) for a description of the *errno* values.

## The extended open operation

An extended open operation is also supported on the device. This operation allows special types of processing during the opening and subsequent closing of the tape device. The **Extended Open** command is

```
tapefd=openx("/dev/rmt0",O_FLAGS,NULL,E_FLAGS);
smcfd=openx("/dev/smc0",O_FLAGS,NULL,E_FLAGS);
```

The O\_FLAGS parameter provides the same options that are described in [Opening the special file for I/O](#). The third parameter is always NULL. The E\_FLAGS parameter provides the extended options. The E\_FLAGS values can be combined during an open operation or they can be used with an OR operation.

The E\_FLAGS parameter has the following flags.

- SC\_RETAIN\_RESERVATION  
This flag prevents the **SCSI Release** command from being sent during a close operation.
- SC\_FORCED\_OPEN  
The flag forces the release of any current reservation on the device by an initiator. The reservation can either be a **SCSI Reserve** or **SCSI Persistent Reserve**.
- SC\_KILL\_OPEN  
This flag kills all currently open processes and then exits the open with *errno* EINPROGRESS returned.
- SC\_PR\_SHARED\_REGISTER  
This flag overrides the configuration reservation type attribute whether it was set to reserve\_6 or persistent. It sets the device driver to use **Persistent Reserve** while the device is open until closed. The configuration reservation type attribute is not changed and the next open without using this flag uses the configuration reservation type. The device driver also registers the host reservation key on the device. This flag can be used with the other extended flags.
- SC\_DIAGNOSTIC  
The device is opened in diagnostic mode, and no SCSI commands are sent to the device during an open operation or a close operation. All operations (such as reserve and mode select) must be processed by the application.

- **SC\_NO\_RESERVE**  
This flag prevents the **SCSI Reserve** command from being sent during an open operation.
- **SC\_PASSTHRU**  
No SCSI commands are sent to the device during an open operation or a close operation. All operations (such as reserve the device, release the device, and set the tape parameters) must be processed explicitly by the application. This flag is the same as the **SC\_DIAGNOSTIC** flag. The exception is that a **SCSI Test Unit Unit Ready** command is issued to the device during an open operation to clear any unit attentions.
- **SC\_FEL**  
This flag turns on the forced error logging in the tape device for read and write operations.
- **SC\_NO\_ERRORLOG**  
This flag turns off the AIX® error logging for all read, write, or IOCTL operations.
- **SC\_TMCP**  
This flag allows up to eight processes to concurrently open a device when the device is already open by another process. There is no restriction for medium changer IOCTL commands that can be issued when this flag is used. However, for tape devices only a limited set of IOCTL commands can be issued. If an IOCTL command cannot be used with this flag, then *errno* EINVAL is returned.

If another process already has the device open with this flag, the open fails, and the *errno* is set to EAGAIN.

If the open system call fails, the *errno* value contains the error code. See [Return codes](#) for a description of the *errno* values.

---

## Writing to the special file

Several subroutines allow writing data to a tape. The basic **write** command is

```
count=write(tapefd, buffer, numbytes);
```

The write operation returns the number of bytes written during the operation. It can be less than the value in numbytes. If the block size is fixed (block\_size≠0), the numbytes value must be a multiple of the block size. If the block size is variable, the value that is specified in numbytes is written. If the count is less than zero, the *errno* value contains the error code that is returned from the driver.

See [Return codes](#) for a description of the *errno* values.

The writev, writex, and writevx subroutines are also supported. Any values that are passed in the ext field with the extended write operation are ignored.

---

## Reading from the special file

Several subroutines allow reading data from a tape. The basic read command is

```
count=read(tapefd, buffer, numbytes);
```

The read operation returns the number of bytes read during the operation. It can be less than the value in numbytes. If the block size is fixed (block\_size≠0), the numbytes value must be a multiple of the block size. If the count is less than zero, the *errno* value contains the error code that is returned from the driver.

See [Return codes](#) for a description of the *errno* values.

If the block size is variable, then the value that is specified in numbytes is read. If the blocks read are smaller than requested, the block is returned up to the maximum size of one block. If the blocks read are greater than requested, an error occurs with the error set to ENOMEM.

Reading a filemark returns a value of zero and positions the tape after the filemark. Continuous reading (after EOM is reached) results in a value of zero and no further change in the tape position.

The readv subroutine is also supported.

---

## Reading with the TAPE\_SHORT\_READ extended parameter

For normal read operations, if the block size is set to variable (0) and the amount of data in a block on the tape is more than the number of bytes requested in the call, an ENOMEM error is returned. An application can read fewer bytes without an error by using the readx or readvx subroutine and specifying the TAPE\_SHORT\_READ extended parameter.

```
count=readx(tapefd, buffer, numbytes, TAPE_SHORT_READ);
```

The TAPE\_SHORT\_READ parameter is defined in the /usr/include/sys/tape.h header file.

---

## Reading with the TAPE\_READ\_REVERSE extended parameter

The TAPE\_READ\_REVERSE extended read parameter reads data from the tape in the reverse direction. The order of the data that is returned in the buffer for each block that is read from the tape is the same as if it were read in the forward direction. However, the last block that is written is the first block in the buffer. This parameter can be used with both fixed and variable block sizes. The TAPE\_SHORT\_READ extended parameter can be used with this parameter, if the block size is set to variable (0).

Use this parameter with the readx or readvx subroutine that specifies the TAPE\_READ\_REVERSE extended parameter.

```
count=readx(tapefd, buffer, numbytes, TAPE_READ_REVERSE);
```

---

## Closing the special file

Closing a special file is a simple process. The file descriptor that is returned by the **Open** command is used to close the command.

```
rc=close(tapefd);
rc=close(smcfd);
```

The return code from the close operation must be checked by the application. If the return code is not zero, the *errno* value is set during a close operation to indicate that a problem occurred while the special file was closing. The close subroutine tries to run as many operations as possible even if there are failures during portions of the close operation. If the device driver cannot terminate the file correctly with filemarks, it tries to close the connection. If the close operation fails, consider the device closed and try another open operation to continue processing the tape. After a close failure, assume that either the data or the tape is inconsistent.

For tape drives, the result of a close operation depends on the special file that was used during the open operation and the tape operation that was run while it was opened. The SCSI commands are issued according to the following logic.

```
If the last tape operation was a WRITE command
 Write 2 filemarks on tape
 If special file is Rewind on Close (Example: /dev/rmt0)
 Rewind tape
 If special file is a No-Rewind on Close (Example: /dev/rmt0.1)
 Backward space 1 filemark (tape is positioned to append next file)

If the last tape operation was a WRITE FILEMARK command
 Write 1 filemark on tape
 If special file is Rewind on Close (Example: /dev/rmt0)
 Rewind tape
 If special file is a No-Rewind on Close (Example: /dev/rmt0.1)
 Backward space 1 filemark (tape is positioned to append next file)

If the last tape operation was a READ command
 If special file is Rewind on Close (Example: /dev/rmt0)
 Rewind tape
 If special file is a No-Rewind on Close (Example: /dev/rmt0.1)
 Forward space to next filemark (tape is positioned to read or append next file)

If the last tape operation was NOT a READ, WRITE, or WRITE FILEMARK command
 If special file is Rewind on Close (Example: /dev/rmt0)
 Rewind tape
 If special file is a No-Rewind on Close (Example: /dev/rmt0.1)
 No commands are issued, tape remains at the current position
```

---

## Device and volume information logging

The device driver provides a logging facility that saves information about the device and the media. The information is extensive for some devices and limited for other devices. If this feature is set to On, either by configuration or the **STIOCSETP** IOCTL, the device driver logging facility gathers all available information through the **SCSI Log Sense** command.

This process is separate 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 the *IBM® TotalStorage™ and System Storage® Tape Device Drivers: Installation and User's Guide* for a description of these parameters.

Each time an **Unload** command or the **STIOC\_LOG\_SENSE** IOCTL command is issued, the log sense data is collected, and an entry is added to the log. Each time a new cartridge is loaded, the log sense data in the tape device is reset so that the log data is gathered on a per-volume basis.

- [Log file](#)

---

## Log file

The data is logged in the /usr/adm/ras directory. The file name is dependent on each device so 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 that are supported by the device. Page 0x00 is followed by all the pages that are specified in page 0x00. The format of each following page is defined in the SCSI specification and the device manual.

The format of the file is defined by the data structure. The logfile\_header is followed by max\_log\_size (or a fewer number of entries for each file). The log\_record\_header is followed by a log entry.

The data structure for log recording is

```

struct logfile_header
{
 char owner[16]; /* module that created the file */
 time_t when; /* time when file created */
 unsigned long count; /* number of entries in file */
 unsigned long first; /* first entry number in wrap queue */
 unsigned long max; /* maximum entries allowed before wrap */
 unsigned long size; /* size of entry (bytes), entry size is fixed */
};

struct log_record_header
{
 time_t when; /* time when log entry made */
 ushort type; /* log entry type */
 #define LOGDEMOUNT 1 /* demount log entry */
 #define LOGSENSE 2 /* log sense ioctl entry */
 #define LOGOVERFLOW 3 /* log overflow entry */
 char device_type[8]; /* device type that made entry */
 char volid[16]; /* volume ID of entry */
 char serial[12]; /* serial number of device */
 reserved[12];
};

```

The format of the log file is

|                   |
|-------------------|
| logfile_header    |
| log_record_header |
| log_record_entry  |
| .                 |
| .                 |
| .                 |
| .                 |
| log_record_header |
| log_record_entry  |

Each log\_record\_entry contains multiple log sense pages. The log pages are placed in order one after another. Each log page contains a header followed by the page contents.

The data structure for the header of the log page is

```

struct log_page_header
{
 char code; /* page code */
 char res; /* reserved */
 unsigned short len; /* length of data in page after header */
};

```

---

## Persistent reservation support and IOCTL operations

- [ODM attributes and configuring persistent reserve support](#)
- [Default device driver host reservation key](#)
- [Preempting and clearing another host reservation](#)
- [Openx\(\) extended parameters](#)
- [AIX tape persistent reserve IOCTLs](#)
- [Atape persistent reserve IOCTLs](#)

---

## ODM attributes and configuring persistent reserve support

Two new ODM attributes are added for PR (Persistent Reservation) support:

- reserve\_type
- reserve\_key

The reserve\_type attribute determines the type of reservation that the device driver uses for the device. The values can be reserve\_6, which is the default for the device driver or persistent. This attribute can be set by either using the AIX SMIT menu to Change/Show Characteristics of a Tape Drive or from a command line with the AIX® command

```
chdev -l rmtx -a reserve_type=persistent or -a reserve_type=reserve_6
```

The reserve\_key attribute is used to optionally set a user-defined host reservation key for the device when the reserve\_type is set to persistent. The default for this attribute is blank (NULL). The default uses a device driver unique host reservation key that is generated for the device. This attribute can be set by either using the AIX SMIT menu to Change/Show Characteristics of a Tape Drive or from a command line with the AIX command

```
chdev -l rmtx -a reserve_key=key
```

The key value can be specified as a 1-8 character ASCII alphanumeric key or a 1-16 hexadecimal key that has the format 0xkey. If fewer than 8 characters are used for an ASCII key such as hostA, the remaining characters are set to 0x00 (NULL).

Note: If the Data Path Failover (DPF) feature is enabled for a logical device by setting the alternate\_pathing attribute to yes, the configuration reserve\_type attribute is not used and the device driver uses persistent reservation. Either the user-defined reserve\_key value or if not defined the default device driver host reservation key is used.

---

## Default device driver host reservation key



If a user-defined host reservation key is not specified, then the device driver uses a unique static host reservation key for the device. This key is generated when the first device is configured and the device driver is initially loaded into kernel memory. The key is 16 hexadecimal digits in the format `0xAppppppppssssssss`, where `pppppp` is the configuration process id that loaded the device driver. Also, `ssssssss` is the 32-bit value of the TOD clock when the device driver was loaded. When any device is configured and the `reserve_key` value is NULL, then the device driver sets the `reserve_key` value to this default internally for the device.

---

## Preempting and clearing another host reservation

When another host initiator is no longer using the device but has left either an SCSI-2 Reserve 6 or a Persistent Reserve active preventing by using the device, either type of reservation can be cleared by using the `openx()` extended parameter `SC_FORCED_OPEN`.

Note: This parameter must be used only when the application or user is sure that the reservation must be cleared.

---

## Openx() extended parameters

The following `openx()` extended parameters are provided for managing device driver reserve during open processing and release during close processing. These parameters apply to either SCSI-2 Reserve 6 or Persistent Reserve. The `SC_PASSTHRU` parameter applies only to the Atape device driver and is defined in `/usr/include/sys/Atape.h`. All other parameters are AIX® system parameters that are defined in `/usr/include/sys/scsi.h`. AIX base tape device drivers might not support all of these parameters.

- `SC_PASSTHRU`
- `SC_DIAGNOSTIC`
- `SC_NO_RESERVE`
- `SC_RETAIN_RESERVATION`
- `SC_PR_SHARED_REGISTER`
- `SC_FORCED_OPEN`

The `SC_PASSTHRU` parameter bypasses all commands that are normally issued on open and close by the device driver. In addition to bypassing the device driver that reserves on open and releases the device on close, all other open commands except test unit ready such as mode selects and rewind on close (if applicable) are also bypassed. A test unit ready is still issued on open to clear any pending unit attentions from the device. This is the only difference in using the `SC_DIAGNOSTIC` parameter.

The `SC_DIAGNOSTIC` parameter bypasses all commands that are normally issued on open and close by the device driver. In addition to bypassing the device driver that reserves on open and releases the device on close, all other open commands such test unit ready, mode selects, and rewind on close (if applicable) are also bypassed.

The `SC_NO_RESERVE` parameter bypasses the device driver that issues a reserve on open only. All other normal open device driver commands are still issued such as test unit ready and mode selects.

The `SC_RETAIN_RESERVATION` parameter bypasses the device driver that issues a release on close only. All other normal close device driver commands are still issued such as rewind (if applicable).

The `SC_PR_SHARED_REGISTER` parameter sets the device driver `reserve_type` to persistent and overrides the configuration `reserve_type` attribute whether it was set to `reserve_6` or persistent. A subsequent reserve on the current open by the device driver (if applicable) uses Persistent Reserve. The `reserve_type` is only changed for the current open. The next open without using this parameter uses the configuration `reserve_type`. In addition to setting the `reserve_type` to persistent, the device driver registers the host reservation key on the device. This parameter can also be used with the extended parameters.

The `SC_FORCED_OPEN` parameter first clears either a SCSI-2 Reserve 6 or a Persistent Reservation if one currently exists on the device from another host. The device driver open processing then continues according to the type of open. This parameter can also be used with the extended parameters.

---

## AIX tape persistent reserve IOCTLs

The Atape device driver supports the AIX® common tape **Persistent Reserve** IOCTLs for application programs to manage their own Persistent Reserve support. The IOCTLs are defined in the header file `/usr/include/sys/tape.h`.

The following two IOCTLs return Persistent Reserve information by using the **SCSI Persistent Reserve In** command.

- **STPRES\_READKEYS**
- **STPRES\_READRES**

The following four IOCTLs complete Persistent Reserve functions by using the **SCSI Persistent Reserve Out** command.

- **STPRES\_CLEAR**
- **STPRES\_PREEMPT**
- **STPRES\_PREEMPT\_ABORT**
- **STPRES\_REGISTER**

Except for the **STPRES\_REGISTER** IOCTL, the other three IOCTLs require that the host reservation key is registered on the device first. This action can be done by either issuing the **STPRES\_REGISTER** IOCTL before the IOCTLs are issued or by opening the device with the `SC_PR_SHARED_REGISTER` parameter.

The **STPRES\_READKEYS** IOCTL issues the **persistent reserve in** command with the read keys service action. The following structure is the argument for this IOCTL.

```
struct st_pres_in {
 ushort version;
 ushort allocation_length;
 uint generation;
 ushort returned_length;
 uchar scsi_status;
 uchar sense_key;
 uchar scsi_asc;
 uchar scsi_ascq;
```

```

 uchar *reservation_info;
}

```

The `allocation_length` is the maximum number of bytes of key values that are returned in the `reservation_info` buffer. The `returned_length` value indicates how many bytes of key values that device reported in the parameter data. Also, it shows the list of key values that are returned by the device up to `allocation_length` bytes. If the `returned_length` is greater than the `allocation_length`, then the application did not provide an `allocation_length` large enough for all of the keys the device registered. The device driver does not consider it an error.

The **STPRES\_READRES** IOCTL issues the **persistent reserve in** command with the read reservations service action. The **STPRES\_READRES** IOCTL uses the same following IOCTL structure as the **STPRES\_READKEYS** IOCTL.

```

struct st_pres_in {
 ushort version;
 ushort allocation_length;
 uint generation;
 ushort returned_length;
 uchar scsi_status;
 uchar sense_key;
 uchar scsi_asc;
 uchar scsi_ascq;
 uchar *reservation_info;
}

```

The allocation length is the maximum number of bytes of reservation descriptors that are returned in the reservation info buffer. The `returned_length` value indicates how many bytes of reservation descriptor values that device reported in the parameter data. Also, it shows the list of reservation descriptor values that are returned by the device up to `allocation_length` bytes. If the `returned_length` is greater than the `allocation_length`, then the application did not provide an `allocation_length` large enough for all of the reservation descriptors the device registered. The device driver does not consider it an error.

The **STPRES\_CLEAR** IOCTL issues the **persistent reserve out** command with the clear service action. The following structure is the argument for this IOCTL.

```

struct st_pres_clear {
 ushort version;
 uchar scsi_status;
 uchar sense_key;
 uchar scsi_asc;
 uchar scsi_ascq;
}

```

The **STPRES\_CLEAR** IOCTL clears a persistent reservation and all persistent reservation registrations on the device.

The **STPRES\_PREEMPT** IOCTL issues the **persistent reserve out** command with the preempt service action. The following structure is the argument for this IOCTL.

```

struct st_pres_preempt {
 ushort version;
 unsigned long long preempt_key;
 uchar scsi_status;
 uchar sense_key;
 uchar scsi_asc;
 uchar scsi_ascq;
}

```

The **STPRES\_PREEMPT** IOCTL preempts a persistent reservation or registration. The `preempt_key` contains the value of the registration key of the initiator that is to be preempted. The determination of whether it is the persistent reservation or registration that is preempted is made by the device. If the initiator corresponding to the `preempt_key` is associated with the reservation that is preempted, then the reservation is preempted and any matching registrations are removed. If the initiator corresponding to the `preempt_key` is not associated with the reservation that is preempted, then any matching registrations are removed. The SPC2 standard states that if a valid request for a preempt service action fails, it can be because of the condition in which another initiator has left the device. The suggested recourse in this case is for the preempting initiator to issue a logical unit reset and retry the preempting service action.

The **STPRES\_PREEMPT\_ABORT** IOCTL issues the **persistent reserve out** command with the preempt and abort service action. The **STPRES\_PREEMPT\_ABORT** IOCTL uses the same argument structure as the **STPRES\_PREEMPT** IOCTL.

```

struct st_pres_preempt {
 ushort version;
 unsigned long long preempt_key;
 uchar scsi_status;
 uchar sense_key;
 uchar scsi_asc;
 uchar scsi_ascq;
}

```

The **STPRES\_PREEMPT\_ABORT** IOCTL preempts a persistent reservation or registration and abort all outstanding commands from the initiators corresponding to the `preempt_key` registration key value. The `preempt_key` contains the value of the registration key of the initiator for which the preempt and abort is to apply. The determination of whether it is the persistent reservation or registration that is to be preempted is made by the device. If the initiator corresponding to the `preempt_key` is associated with the reservation that is preempted, then the reservation is preempted and any matching registrations are removed. If the initiator corresponding to the `preempt_key` is not associated with the reservation that is preempted, then any matching registrations are removed. Regardless of whether the preempted initiator holds the reservation, all outstanding commands from all initiators corresponding to the `preempt_key` are aborted.

The **STPRES\_REGISTER** IOCTL issues the **persistent reserve out** command with the register service action. The following structure is the argument for this IOCTL.

```

struct st_pres_register {
 ushort version;
 uchar scsi_status;
 uchar sense_key;
 uchar scsi_asc;
 uchar scsi_ascq;
}

```

The **STPRES\_REGISTER** IOCTL registers the current host persistent reserve registration key value with the device. The **STPRES\_REGISTER** IOCTL is only supported if the device is opened with a `reserve_type` set to persistent, otherwise an error of `EACCESS` is returned. The intended use of this IOCTL is to allow a preempted host to regain access to a shared device without requiring that the device is closed and reopened.

If a persistent reserve IOCTL fails, the return code is set to -1 and the *errno* value is set to one of the following.

- ENOMEM Device driver cannot obtain memory to run the command.
- EFAULT An error occurred while the caller's data buffer was manipulated
- EACCES The device is opened with a *reserve\_type* set to *reserve\_6*
- EINVAL The requested IOCTL is not supported by this version of the device driver or invalid parameter that is provided in the argument structure
- ENXIO The device indicated that the persistent reserve command is not supported
- EBUSY The device returned a SCSI status byte of RESERVATION CONFLICT or BUSY. Or, the reservation for the device was preempted by another host and the device driver does not issue further commands.
- EIO Unknown I/O failure occurred on the command

---

## Atape persistent reserve IOCTLs

The Atape device driver provides **Persistent Reserve** IOCTLs for application programs to manage their own Persistent Reserve support. These IOCTLs are defined in the header file */usr/include/sys/Atape\_pr.h*.

The following IOCTLs return Persistent Reserve information by using the **SCSI Persistent Reserve In** command.

- **STIOC\_READ\_RESERVEKEYS**
- **STIOC\_READ\_RESERVATIONS**
- **STIOC\_READ\_RESERVE\_FULL\_STATUS**

The following IOCTLs complete Persistent Reserve functions by using the **SCSI Persistent Reserve Out** command.

- **STIOC\_REGISTER\_KEY**
- **STIOC\_REMOVE\_REGISTRATION**
- **STIOC\_CLEAR\_ALL\_REGISTRATIONS**
- **STIOC\_PREEMPT\_RESERVATION**
- **STIOC\_PREEMPT\_ABORT**
- **STIOC\_CREATE\_PERSISTENT\_RESERVE**

The following IOCTLs are modified to handle both SCSI-2 Reserve 6 and Persistent Reserve based on the current *reserve\_type* setting.

- **SIOC\_RESERVE**
- **SIOC\_RELEASE**

The **STIOC\_READ\_RESERVEKEYS** IOCTL returns the reservation keys from the device. The argument for this IOCTL is the address of a *read\_keys* structure. If the *reserve\_key\_list* pointer is NULL, then only the generation and length fields are returned. This action allows an application to first obtain the length of the *reserve\_key\_list* and malloc a return buffer before the IOCTL is issued with a *reserve\_key\_list* pointer to that buffer. If the return length is 0, then no reservation keys are registered with the device.

The following structure is used for this IOCTL.

```
struct read_keys
{
 uint generation; /* counter for PERSISTENT RESERVE OUT requests */
 uint length; /* number of bytes in the Reservation Key list */
 ullong *reserve_key_list; /* list of reservation keys */
};
```

The **STIOC\_READ\_RESERVATIONS** IOCTL returns the current reservations from the device if any exist. The argument for this IOCTL is the address of a *read\_reserves* structure. If the *reserve\_list* pointer is NULL, then only the generation and length fields are returned. This action allows an application to first obtain the length of the *reserve\_list* and malloc a return buffer before the IOCTL is issued with a *reserve\_list* pointer to that buffer. If the return length is 0, then no reservations currently exist on the device.

The following structures are used for this IOCTL.

```
struct reserve_descriptor
{
 ullong key; /* reservation key */
 uint scope_spec_addr; /* scope-specific address */
 uchar reserved;
 uint scope:4; /* persistent reservation scope */
 uint type:4; /* reservation type */
 ushort ext_length; /* extent length */
};

struct read_reserves
{
 uint generation; /* counter for PERSISTENT RESERVE OUT requests */
 uint length; /* number of bytes in the Reservation list */
 struct reserve_descriptor* reserve_list; /* list of reservation key descriptors */
};
```

The **STIOC\_READ\_RESERVE\_FULL\_STATUS** IOCTL returns extended information for all reservation keys and reservations from the device if any exist. The argument for this IOCTL is the address of a *read\_full\_status* structure. If the *status\_list* pointer is NULL, then only the generation and length fields are returned. This action allows an application to first obtain the length of the *status\_list* and malloc a return buffer before the IOCTL is issued with a *status\_list* pointer to that buffer. If the return length is 0, then no reservation keys or reservations currently exist on the device.

The following structures are used for this IOCTL.

```
struct transport_id
{
 uint format_code:2,
 rsvd:2,
```

```

 protocol_id:4;
};

struct fcp_transport_id
{
 uint format_code:2,
 rsvd:2,
 protocol_id:4;
 char reserved1[7];
 ullong n_port_name;
 char reserved2[8];
};

struct scsi_transport_id
{
 uint format_code:2,
 rsvd:2,
 protocol_id:4;
 char reserved1[1];
 ushort scsi_address;
 ushort obsolete;
 ushort target_port_id;
 char reserved2[16];
};

struct sas_transport_id
{
 uint format_code:2,
 rsvd:2,
 protocol_id:4;
 char reserved1[3];
 ullong sas_address;
 char reserved2[12];
};

struct status_descriptor
{
 ullong key; /* reservation key */
 char reserved1[4];
 uint rsvd:5,
 spc2_r:1, /* future use for SCSI-2 reserve */
 all_tg_pt:1, /* all target ports */
 r_holder:1, /* reservation holder */
 scope:4, /* persistent reservation scope */
 type:4; /* reservation type */
 char reserved2[4];
 ushort target_port_id; /* relative target port id */
 uint descriptor_length; /* additional descriptor length */
 union {
 struct transport_id transport_id; /* transport ID */
 struct fcp_transport_id fcp_id; /* FCP transport ID */
 struct sas_transport_id sas_id; /* SAS transport ID */
 struct scsi_transport_id scsi_id; /* SCSI transport ID */
 };
};

struct read_full_status
{
 uint generation; /* counter for PERSISTENT RESERVE OUT requests */
 uint length; /* number of bytes for total status descriptors */
 struct status_descriptor *status_list; /* list of reserve status descriptors */
};

```

The **STIOC\_REGISTER\_KEY** IOCTL registers a host reservation key on the device. The argument for this IOCTL is the address of an unsigned long key that can be 1 - 16 hexadecimal digits. If the key value is 0, then the device driver registers the configuration reserve key on the device. This key is either a user-specified host key or the device driver default host key.

If the host has a current persistent reservation on the device and the key is different from the current reservation key, the reservation is retained and the host reservation key is changed to the new key.

The **STIOC\_REMOVE\_REGISTRATION** IOCTL removes the host reservation key and reservation if one exists from the device. There is no argument for this IOCTL. The **SIIOC\_RELEASE** IOCTL can also be used to complete the same function.

The **STIOC\_CLEAR\_ALL\_REGISTRATIONS** IOCTL clears all reservation keys and reservations on the device (if any exist) for the same host and any other host. There is no argument for this IOCTL.

The **STIOC\_PREEMPT\_RESERVATION** IOCTL registers a host reservation key on the device and then preempts the reservation that is held by another host if one exists. Or, it creates a new persistent reservation by using the host reservation key. The argument for this IOCTL is the address of an unsigned long key that can be 1 - 16 hexadecimal digits. If the key value is 0, then the device driver registers the configuration reserve key on the device. This key is either a user-specified host key or the device driver default host key.

The **STIOC\_PREEMPT\_ABORT** IOCTL registers a host reservation key on the device, preempts the reservation that is held by another host, and clears the task that is set for the preempted initiator if one exists. Or, it creates a new persistent reservation by using the host reservation key. The argument for this IOCTL is the address of an unsigned long key that can be 1 - 16 hexadecimal digits. If the key value is 0, then the device driver registers the configuration reserve key on the device. This key is either a user-specified host key or the device driver default host key.

The **STIOC\_CREATE\_PERSISTENT\_RESERVE** IOCTL creates a persistent reservation on the device by using the host reservation key that was registered with the **STIOC\_REGISTER\_KEY** IOCTL. There is no argument for this IOCTL. The **SIIOC\_RESERVE** IOCTL can also be used to complete the same function.

The **SIIOC\_RESERVE** IOCTL reserves the device. If the `reserve_type` is set to `reserve_6`, the device driver issues a **SCSI Reserve 6** command. If the `reserve_type` is set to persistent, the device driver first registers the current host reservation key and then creates a persistent reservation. The current host reservation key can be either the configuration key for the device or a key that was registered previously with the **STIOC\_REGISTER\_KEY** IOCTL.

The **STIOC\_RELEASE** IOCTL releases the device. If the `reserve_type` is set to `reserve_6`, the device driver issues a **SCSI Release 6** command. If the `reserve_type` is set to persistent, the device driver removes the host reservation key and reservation if one exists from the device.

If a persistent reserve IOCTL fails, the return code is set to -1 and the `errno` value is set to one of the following.

- ENOMEM Device driver cannot obtain memory to complete the command.
- EFAULT An error occurred while the caller's data buffer was manipulated
- EACCES The current open is using a `reserve_type` set to `reserve_6`
- EINVAL Device does not support either the **SCSI Persistent Reserve In/Out** command, the service action for the command, or the sequence of the command such as issuing the **STIOC\_REMOVE\_REGISTRATION** IOCTL when no reservation key was registered for the host.
- EBUSY Device failed the command with reservation conflict. Either a SCSI-2 Reserve 6 reservation is active, the sequence of the command such as issuing the **STIOC\_CREATE\_PERSISTENT\_RESERVE** IOCTL when no reservation key was registered for the host, or the reservation for the device was preempted by another host and the device driver does not issue further commands.
- EIO Unknown I/O failure occurred on the command.

---

## General IOCTL operations

This chapter describes the IOCTL commands that provide control and access to the tape and medium changer devices. These commands are available for all tape and medium changer devices. They can be issued to any `rmt*`, `rmt*.smc`, or `smc*` special file.

- [Overview](#)

---

## Overview

The following IOCTL commands are supported.

**IOCINFO**  
Return device information.

**STIOCMD**  
Issue the **AIX Pass-through** command.

**STPASSTHRU**  
Issue the **AIX Pass-through** command.

**SIOC\_PASSTHRU\_COMMAND**  
Issue the **Atape Pass-through** command.

**SIOC\_INQUIRY**  
Return inquiry data.

**SIOC\_REQSENSE**  
Return sense data.

**SIOC\_RESERVE**  
Reserve the device.

**SIOC\_RELEASE**  
Release the device.

**SIOC\_TEST\_UNIT\_READY**  
Issue a **SCSI Test Unit Ready** command.

**SIOC\_LOG\_SENSE\_PAGE**  
Return log sense data for a specific page.

**SIOC\_LOG\_SENSE10\_PAGE**  
Return log sense data for a specific page and Subpage.

**SIOC\_MODE\_SENSE\_PAGE**  
Return mode sense data for a specific page.

**SIOC\_MODE\_SENSE\_SUBPAGE**  
Return mode sense data for a specific page and subpage.

**SIOC\_MODE\_SENSE**  
Return whole mode sense data include header, block descriptor, and page for a specific page.

**SIOC\_MODE\_SELECT\_PAGE**  
Set mode sense data for a specific page.

**SIOC\_MODE\_SELECT\_SUBPAGE**  
Set mode sense data for a specific page and subpage.

**SIOC\_INQUIRY\_PAGE**  
Return inquiry data for a specific page.

**SIOC\_DISABLE\_PATH**  
Manually disable (fence) a SCSI path for a device.

**SIOC\_ENABLE\_PATH**  
Enable a manually disabled (fenced) SCSI path for a device.

**SIOC\_SET\_PATH**  
Explicitly set the current path that is used by the device driver.

**SIOC\_QUERY\_PATH**  
Query device and path information for the primary and first alternate SCSI path for a device. This IOCTL is obsolete but still supported. The **SIOC\_DEVICE\_PATHS** IOCTL can be used instead of this IOCTL.

**SIOC\_DEVICE\_PATHS**  
Query device and path information for the primary and all alternate SCSI paths for the device.

**SIOC\_RESET\_PATH**  
Issue an **Inquiry** command on each SCSI path that is not manually disabled (fenced) and enable the path if the **Inquiry** command succeeds.

**SIOC\_CHECK\_PATH**  
Completes the same function as the **SIOC\_RESET\_PATH** IOCTL.

**SIOC\_QUERY\_OPEN**  
Returns the process ID that currently has the device opened.

**SIOC\_RESET\_DEVICE**  
Issues a SCSI target reset or SCSI lun reset (for FCP or SAS attached) to the device.

**SIOC\_DRIVER\_INFO**  
Query the device driver information.

These IOCTL commands and their associated structures are defined by including the `/usr/include/sys/Atape.h` header file in the C program by using the functions.

- [IOCINFO](#)
- [STIOCMD](#)
- [STPASSTHRU](#)
- [SIOC PASSTHRU COMMAND](#)
- [SIOC INQUIRY](#)
- [SIOC REQSENSE](#)
- [SIOC RESERVE](#)
- [SIOC RELEASE](#)
- [SIOC TEST UNIT READY](#)
- [SIOC LOG SENSE PAGE](#)
- [SIOC LOG SENSE10 PAGE](#)
- [SIOC MODE SENSE PAGE](#)
- [SIOC MODE SENSE SUBPAGE](#)
- [SIOC MODE SELECT PAGE](#)
- [SIOC MODE SELECT SUBPAGE](#)
- [SIOC QUERY OPEN](#)
- [SIOC INQUIRY PAGE](#)
- [SIOC DISABLE PATH](#)
- [SIOC ENABLE PATH](#)
- [SIOC SET PATH](#)
- [SIOC DEVICE PATHS](#)
- [SIOC QUERY PATH](#)
- [SIOC RESET PATH and SIOC CHECK PATH](#)
- [SIOC RESET DEVICE](#)
- [SIOC DRIVER INFO](#)

---

## IOCINFO

This IOCTL command provides access to information about the tape or medium changer device. It is a standard AIX® IOCTL function.

An example of the **IOCINFO** command is

```
#include <sys/devinfo.h>
#include <sys/Atape.h>
struct devinfo info;

if (!ioctl (fd, IOCINFO, &info))
{
 printf ("The IOCINFO ioctl succeeded\n");
}
else
{
 perror ("The IOCINFO ioctl failed");
}
```

An example of the output data structure for a tape drive `rmt*` special file is

```
info.devtype=DD_SCTAPE
info.devsubtype=ATAPE_3590
info.un.scmt.type=DT_STREAM
info.un.scmt.blksize=tape block size (0=variable)
```

An example of the output data structure for an integrated medium changer `rmt*.smc` special file is

```
info.devtype=DD_MEDIUM_CHANGER;
info.devsubtype=ATAPE_3590;
```

An example of the output data structure for an independent medium changer `smc*` special file is

```
info.devtype=DD_MEDIUM_CHANGER;
info.devsubtype=ATAPE_7337;
```

See the `Atape.h` header file for the defined `devsubtype` values.

---

## STIOCMD

This IOCTL command issues the **SCSI Pass-through** command. It is used by the diagnostic and service aid routines. The structure for this command is in the `/usr/include/sys/scsi.h` file.

This IOCTL is supported on both SCSI adapter attached devices and FCP adapter attached devices. For FCP adapter devices, the returned `adapter_status` field is converted from the FCP codes that are defined in `/usr/include/sys/scsi_buf.h` to the SCSI codes defined in `/usr/include/sys/scsi.h`, if possible. This action is to provide downward compatibility with existing applications that use the **STIOCMD** IOCTL for SCSI attached devices.

Note: There is no interaction by the device driver with this command. The error handling and logging functions are disabled. If the command results in a check condition, the application must issue a **Request Sense** command to clear any contingent allegiance with the device.

An example of the **STIOCMD** command is

```
struct sc_iocmd sciocmd;
struct inquiry_data inqdata;

bzero(&sciocmd, sizeof(struct sc_iocmd));
bzero(&inqdata, sizeof(struct inquiry_data));

/* issue inquiry */
sciocmd.scsi_cdb[0]=0x12;
sciocmd.timeout_value=200; /* SECONDS */
sciocmd.command_length=6;
sciocmd.buffer=(char *)&inqdata;
sciocmd.data_length=sizeof(struct inquiry_data);
sciocmd.scsi_cdb[4]=sizeof(struct inquiry_data);
sciocmd.flags=B_READ;

if (!ioctl (sfdd, STIOCMD, &sciocmd))
{
 printf ("The STIOCMD ioctl for Inquiry Data succeeded\n");
 printf ("\nThe inquiry data is:\n");
 dump_bytes (&inqdata, sizeof(struct inquiry_data),"Inquiry Data");
}
else
{
 perror ("The STIOCMD ioctl for Inquiry Data failed");
}
```

---

## STPASSTHRU

This IOCTL command issues the **AIX Pass-through** command that is supported by base AIX® tape device drivers. The IOCTL command and structure are defined in the header files `/usr/include/sys/scsi.h` and `/usr/include/sys/tape.h`. Refer to AIX documentation for information about using the command.

---

## SIOC\_PASSTHRU\_COMMAND

This IOCTL command issues the Atape device driver **Pass-through** command. The data structure that is used on this IOCTL is

```
struct scsi_passthru_cmd {
 uchar command_length; /* Length of SCSI command 6, 10, 12 or 16 */
 uchar scsi_cdb[16]; /* SCSI command descriptor block */
 uint timeout_value; /* Timeout in seconds or 0 for command default */
 uint buffer_length; /* Length of data buffer or 0 */
 char *buffer; /* Pointer to data buffer or NULL */
 uint number_bytes; /* Number of bytes transferred to/from buffer */
 uchar sense_length; /* Number of valid sense bytes */
 uchar sense[MAXSENSE]; /* Sense data when sense length > 0 */
 uint trace_length; /* Number bytes in buffer to trace, 0 for none */
 char read_data_command; /* Input flag, set it to 1 for read type cmds */
 char reserved[27];
};
```

The arg parameter for the IOCTL is the address of a `scsi_passthru_cmd` structure.

The device driver issues the SCSI command by using the `command_length` and `scsi_cdb` fields. If the command receives data from the device (such as SCSI Inquiry), then the application must also set the `buffer_length` and buffer pointer for the return data along with the `read_data_command` set to 1. For commands that send data to the device (such as **SCSI Mode Select**), the `buffer_length` and pointer is set for the send data and the `read_data_command` set to 0. If the command has no data transfer, the buffer length is set to 0 and buffer pointer that is set to NULL.

The specified `timeout_value` field is used if not 0. If 0, then the device driver assigns its internal timeout value that is based on the SCSI command.

The `trace_length` field is normally used only for debug. It specifies the number of bytes on a data transfer type command that is traced when the AIX® Atape device driver trace is running.

If the SCSI command fails, then the IOCTL returns -1 and `errno` value is set for the failing command. If the device returned sense data for the failure, then the `sense_length` is set to the number of sense bytes returned in the sense field. If there was no sense data for the failure, the `sense_length` is 0.

If the SCSI command transfers data either to or from the device, then the `number_bytes` fields indicate how many bytes were transferred.

---

## SIOC\_INQUIRY

This IOCTL command collects the inquiry data from the device.

The data structure is

```
struct inquiry_data
{
 uint qual:3, /* peripheral qualifier */
 uint type:5; /* device type */
 uint rm:1, /* removable medium */
};
```

```

uint mod:7; /* device type modifier */
uint iso:2; /* ISO version */
uint ecma:3; /* ECMA version */
uint ansi:3; /* ANSI version */
uint aenc:1; /* asynchronous event notification */
uint trmiop:1; /* terminate I/O process message */
uint :2; /* reserved */
uint rdf:4; /* response data format */
uchar len; /* additional length */
uchar resvd1; /* reserved */
uint :4; /* reserved */
uint mchngr:1; /* Medium Changer mode (SCSI-3 only) */
uint :3; /* reserved */
uint reladr:1; /* relative addressing */
uint wbus32:1; /* 32-bit wide data transfers */
uint wbus16:1; /* 16-bit wide data transfers */
uint sync:1; /* synchronous data transfers */
uint linked:1; /* linked commands */
uint :1; /* reserved */
uint cmdque:1; /* command queueing */
uint sftre:1; /* soft reset */
uchar vid[8]; /* vendor ID */
uchar pid[16]; /* product ID */
uchar revision[4]; /* product revision level */
uchar vendor1[20]; /* vendor specific */
uchar resvd2[40]; /* reserved */
uchar vendor2[31]; /* vendor specific (padded to 127) */
};

```

An example of the **SIOC\_INQUIRY** command is

```

#include <sys/Atape.h>

struct inquiry_data inquiry_data;

if (!ioctl (fd, SIOC_INQUIRY, &inquiry_data))
{
 printf ("The SIOC_INQUIRY ioctl succeeded\n");
 printf ("\nThe inquiry data is:\n");
 dump_bytes ((uchar *)&inquiry_data, sizeof (struct inquiry_data));
}
else
{
 perror ("The SIOC_INQUIRY ioctl failed");
 sioc_request_sense();
}

```

## SIOC\_REQSENSE

This IOCTL command returns the device sense data. If the last command resulted in an input/output error (EIO), the sense data is returned for the error. Otherwise, a new sense command is issued to the device.

The data structure is

```

struct request_sense
{
 uint valid:1; /* sense data is valid */
 uint err_code:7; /* error code */
 uchar segnum; /* segment number */
 uint fm:1; /* filemark detected */
 uint eom:1; /* end of medium */
 uint ili:1; /* incorrect length indicator */
 uint resvd1:1; /* reserved */
 uint key:4; /* sense key */
 signed int info; /* information bytes */
 uchar addlen; /* additional sense length */
 uint cmdinfo; /* command specific information */
 uchar asc; /* additional sense code */
 uchar ascq; /* additional sense code qualifier */
 uchar fru; /* field replaceable unit code */
 uint sksv:1; /* sense key specific valid */
 uint cd:1; /* control/data */
 uint resvd2:2; /* reserved */
 uint bpv:1; /* bit pointer valid */
 uint sim:3; /* system information message */
 uchar field[2]; /* field pointer */
 uchar vendor[109]; /* vendor specific (padded to 127) */
};

```

An example of the **SIOC\_REQSENSE** command is

```

#include <sys/Atape.h>

struct request_sense sense_data;

if (!ioctl (smcfd, SIOC_REQSENSE, &sense_data))
{
 printf ("The SIOC_REQSENSE ioctl succeeded\n");
 printf ("\nThe request sense data is:\n");
 dump_bytes ((uchar *)&sense_data, sizeof (struct request_sense));
}

```



```

else
{
 perror ("The SIOC_REQSENSE ioctl failed");
}

```

---

## SIOC\_RESERVE

This IOCTL command reserves the device to the device driver. The specific SCSI command that is issued to the device depends on the current reservation type that is used by the device driver, either a SCSI Reserve or Persistent Reserve.

There are no arguments for this IOCTL command.

An example of the **SIOC\_RESERVE** command is

```

#include <sys/Atape.h>

if (!ioctl (fd, SIOC_RESERVE, NULL))
{
 printf ("The SIOC_RESERVE ioctl succeeded\n");
}
else
{
 perror ("The SIOC_RESERVE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_RELEASE

This IOCTL command releases the current device driver reservation on the device. The specific SCSI command that is issued to the device depends on the current reservation type that is used by the device driver, either a SCSI Reserve or Persistent Reserve.

There are no arguments for this IOCTL command.

An example of the **SIOC\_RELEASE** command is

```

#include <sys/Atape.h>

if (!ioctl (fd, SIOC_RELEASE, NULL))
{
 printf ("The SIOC_RELEASE ioctl succeeded\n");
}
else
{
 perror ("The SIOC_RELEASE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_TEST\_UNIT\_READY

This IOCTL command issues the **SCSI Test Unit Ready** command to the device.

There are no arguments for this IOCTL command.

An example of the **SIOC\_TEST\_UNIT\_READY** command is

```

#include <sys/Atape.h>

if (!ioctl (fd, SIOC_TEST_UNIT_READY, NULL))
{
 printf ("The SIOC_TEST_UNIT_READY ioctl succeeded\n");
}
else
{
 perror ("The SIOC_TEST_UNIT_READY ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_LOG\_SENSE\_PAGE

This IOCTL command returns a log sense page from the device. The page is selected by specifying the `page_code` in the `log_sense_page` structure. Optionally, a specific parm pointer, also known as a parm code, and the number of parameter bytes can be specified with the command.

To obtain the entire log page, the `len` and `parm_pointer` fields are set to zero. To obtain the entire log page that starts at a specific parameter code, set the `parm_pointer` field to the wanted code and the `len` field to zero. To obtain a specific number of parameter bytes, set the `parm_pointer` field to the wanted code. Then, set the `len` field to the number of parameter bytes plus the size of the log page header (4 bytes). The first 4 bytes of returned data are always the log page header.

See the appropriate device manual to determine the supported log pages and content.

The data structure is

```
struct log_sense_page
{
 char page_code;
 unsigned short len;
 unsigned short parm_pointer;
 char data[LOGSENSEPAGE];
};
```

An example of the **SIOC\_LOG\_SENSE\_PAGE** command is

```
#include <sys/Atape.h>

struct log_sense_page log_page;
int temp;

/* get log page 0, list of log pages */
log_page.page_code = 0x00;
log_page.len = 0;
log_page.parm_pointer = 0;

if (!ioctl (fd, SIOC_LOG_SENSE_PAGE, &log_page))
{
 printf ("The SIOC_LOG_SENSE_PAGE ioctl succeeded\n");
 dump_bytes(log_page.data, LOGSENSEPAGE);
}
else
{
 perror ("The SIOC_LOG_SENSE_PAGE ioctl failed");
 sioc_request_sense();
}

/* get 3590 fraction of volume traversed */
log_page.page_code = 0x38;
log_page.len = 0;
log_page.parm_pointer = 0x000F;

if (!ioctl (fd, SIOC_LOG_SENSE_PAGE, &log_page))
{
 temp = log_page.data[(sizeof(log_page_header) + 4)];
 printf ("The SIOC_LOG_SENSE_PAGE ioctl succeeded\n");
 printf ("Fractional Part of Volume Traversed %x\n",temp);
}
else
{
 perror ("The SIOC_LOG_SENSE_PAGE ioctl failed");
 sioc_request_sense();
}
```

---

## SIOC\_LOG\_SENSE10\_PAGE

This IOCTL command is enhanced to add a subpage variable from **SIOC\_LOG\_SENSE\_PAGE**. It returns a log sense page or subpage from the device. The page is selected by specifying the `page_code` or `subpage_code` in the `log_sense10_page` structure. Optionally, a specific `parm_pointer`, also known as a `parm code`, and the number of parameter bytes can be specified with the command.

To obtain the entire log page, the `len` and `parm_pointer` fields are set to zero. To obtain the entire log page that starts at a specific parameter code, set the `parm_pointer` field to the wanted code and the `len` field to zero. To obtain a specific number of parameter bytes, set the `parm_pointer` field to the wanted code. Then, set the `len` field to the number of parameter bytes plus the size of the log page header (4 bytes). The first 4 bytes of returned data are always the log page header. See the appropriate device manual to determine the supported log pages and content.

The data structure is

```
/* log sense page and subpage structure */
struct log_sense10_page
{
 uchar page_code; /* [IN] log sense page code */
 uchar subpage_code; /* [IN] log sense Subpage code */
 uchar reserved[2];
 unsigned short len; /* [IN] specific allocation length for the data */
 /* [OUT] number of valid bytes in */
 /* data(log_page_header_size+page_length) */
 unsigned short parm_pointer; /* [IN] specific parameter number at which */
 /* the data begins */
 char data[LOGSENSEPAGE]; /* [OUT] log sense page and Subpage data */
};
```

An example of the **SIOC\_LOG\_SENSE10\_PAGE** command is

```
#include <sys/Atape.h>

struct log_sense10_page logdata10;
struct log_page_header *page_header;
char text[80];

logdata10.page_code = page;
logdata10.subpage_code = subpage;
logdata10.len = len;
```

```

logdata10.parm_pointer = parm;
page_header = (struct log_page_header *)logdata10.data;

printf("Issuing log sense for page 0x%02X and subpage 0x%02X...\n",page,subpage);

if (!ioctl (fd, SIOC_LOG_SENSE10_PAGE, &logdata10))
{
 sprintf(text,"Log Sense Page 0x%02X, Subpage 0x%02X, Page Length %d
Data",page,subpage,logdata10.len);
 dump_bytes(logdata10.data,logdata10.len,text);
}
else
{
 perror ("The SIOC_LOG_SENSE10_PAGE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_MODE\_SENSE\_PAGE

This IOCTL command returns a mode sense page from the device. The page is selected by specifying the page\_code in the mode\_sense\_page structure.

See the appropriate device manual to determine the supported mode pages and content.

The data structure is

```

struct mode_sense_page
{
 char page_code;
 char data[MODESENSEPAGE];
};

```

An example of the **SIOC\_MODE\_SENSE\_PAGE** command is

```

#include <sys/Atape.h>

struct mode_sense_page mode_page;

/* get Medium Changer mode */
mode_page.page_code = 0x20;
if (!ioctl (fd, SIOC_MODE_SENSE_PAGE, &mode_page))
{
 printf ("The SIOC_MODE_SENSE_PAGE ioctl succeeded\n");
 if (mode_page.data[2] == 0x02)
 printf ("The library is in Random mode.\n");
 else
 if (mode_page.data[2] == 0x05)
 printf ("The library is in Automatic (Sequential) mode.\n");
}
else
{
 perror ("The SIOC_MODE_SENSE_PAGE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_MODE\_SENSE\_SUBPAGE

This IOCTL command returns the whole mode sense data, including header, block descriptor, and page code for a specific page or subpage from the device. The wanted page or subpage is inputted by specifying the page\_code and subpage\_code in the mode\_sense structure.

The data structure is

```

struct mode_sense
{
 uchar page_code; /* [IN] mode sense page code */
 uchar subpage_code; /* [IN] mode sense subpage code */
 uchar reserved[6];
 uchar cmd_code; /* [OUT] SCSI Command Code: this field is set with */
 /* SCSI command code which the device responded. */
 /* x'5A' = Mode Sense (10) */
 /* x'1A' = Mode Sense (6) */
 char data[MODESENSEPAGE]; /* [OUT] whole mode sense data include header,
block descriptor and page */
};

```

An example of the **SIOC\_MODE\_SENSE** command is

```

#include <sys/Atape.h>

struct mode_sense modedata;
char text[80];
bzero(&modedata, sizeof(struct mode_sense));
modedata.page_code = page;
modedata.subpage_code = subpage;

printf("Issuing mode sense subpage for page 0x%02X subpage 0x%02X...\n",
page,subpage);

```

```

if (!ioctl (fd, SIOC_MODE_SENSE, &modedata))
{
 sprintf(text,"Mode Sense 0x%02X Subpage 0x%02X cmd_code 0x%02X",
 modedata.page_code,modedata.subpage_code,modedata.cmd_code);
 dump_bytes((char *) &modedata, sizeof(struct mode_sense), text);
}
else
{
 perror ("The SIOC_MODE_SENSE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_MODE\_SELECT\_PAGE

This IOCTL command sets device parameters in a specific mode page. The wanted page is selected by specifying the `page_code` in the `mode_sense_page` structure. See the appropriate device manual to determine the supported mode pages and parameters that can be modified. The `arg` parameter for the IOCTL is the address of a `mode_sense_page` structure.

The data structure is

```

struct mode_sense_page
{
 uchar page_code; /* mode sense page code */
 char data[MODESENSEPAGE];
};

```

This data structure is also used for the **SIOC\_MODE\_SENSE\_PAGE** IOCTL. The application must issue the **SIOC\_MODE\_SENSE\_PAGE** IOCTL, and modify the wanted bytes in the returned `mode_sense_page` structure data field. Then, it issues this IOCTL with the modified fields in the structure.

---

## SIOC\_MODE\_SELECT\_SUBPAGE

This IOCTL command sets device parameters in a specific mode page and subpage. The wanted page and subpage are selected by specifying the `page_code` and `subpage_code` in the `mode_sense_subpage` structure. See the appropriate device manual to determine the supported mode pages, subpages, and parameters that can be modified. The `arg` parameter for the IOCTL is the address of a `mode_sense_subpage` structure.

The data structure is

```

struct mode_sense_subpage
{
 uchar page_code; /* mode sense page code */
 uchar subpage_code; /* mode sense subpage code */
 uint reserved:7,
 sp_bit:1; /* mode select save page bit */
 char data[MODESENSEPAGE];
};

```

This data structure is also used for the **SIOC\_MODE\_SENSE\_SUBPAGE** IOCTL. The application must issue the **SIOC\_MODE\_SENSE\_SUBPAGE** IOCTL, and modify the wanted bytes in the returned `mode_sense_subpage` structure data field. Then, it issues this IOCTL with the modified fields in the structure. If the device supports setting the `sp_bit` for the mode page to 1, then the `sp_bit` field can be set to 0 or 1. If the device does not support the `sp_bit`, then the `sp_bit` field must be set to 0.

---

## SIOC\_QUERY\_OPEN

This IOCTL command returns the ID of the process that currently has a device open. There is no associated data structure. The `arg` parameter specifies the address of an `int` for the return process ID.

If the application opened the device by using the extended open parameter `SC_TMCP`, the process ID is returned for any other process that has the device open currently. Or, zero is returned if the device is not currently open. If the application opened the device without the extended open parameter `SC_TMCP`, the process ID of the current application is returned.

An example of the **SIOC\_QUERY\_OPEN** command is

```

#include <sys/Atape.h>

int sioc_query_open (void)
{
 int pid = 0;

 if (ioctl(fd, SIOC_QUERY_OPEN, &pid) == 0)
 {
 if (pid)
 printf("Device is currently open by process id %d\n",pid)
 else
 printf("Device is not open\n");
 }
 else
 printf("Error querying device open...\n");

 return errno;
}

```

---

## SIOC\_INQUIRY\_PAGE

This IOCTL command returns an inquiry page from the device. The page is selected by specifying the `page_code` in the `inquiry_page` structure.

See the appropriate device manual to determine the supported inquiry pages and content.

The data structure is

```
struct inquiry_page
{
 char page_code;
 char data[INQUIRYPAGE];
};
```

An example of the **SIOC\_INQUIRY\_PAGE** command is

```
#include <sys/Atape.h>

struct inquiry_page inq_page;

/* get inquiry page x83 */
inq_page.page_code = 0x83;
if (!ioctl (fd, SIOC_INQUIRY_PAGE, &inq_page))
{
 printf ("The SIOC_INQUIRY_PAGE ioctl succeeded\n");
}
else
{
 perror ("The SIOC_INQUIRY_PAGE ioctl failed");
 sioc_request_sense();
}
```

---

## SIOC\_DISABLE\_PATH

This IOCTL command manually disables (fences) the device driver from using either the primary or an alternate SCSI path to a device until the **SIOC\_ENABLE\_PATH** command is issued for the same path that is manually disabled. The `arg` parameter on the IOCTL command specifies the path to be disabled. The primary path is path 1, the first alternate path 2, the second alternate path 3, and so on. This command can be used concurrently when the device is already open by another process by using the `openx()` extended parameter `SC_TMCP`.

This IOCTL command is valid only if the device has one or more alternate paths configured. Otherwise, the IOCTL command fails with *errno* set to `EINVAL`. The **SIOC\_DEVICE\_PATHS** IOCTL command can be used to determine the paths that are enabled or manually disabled.

An example of the **SIOC\_DISABLE\_PATH** command is

```
#include <sys/Atape.h>

/* Disable primary SCSI path */
ioctl(fd, SIOC_DISABLE_PATH, PRIMARY_SCSI_PATH);

/* Disable alternate SCSI path */
ioctl(fd, SIOC_DISABLE_PATH, ALTERNATE_SCSI_PATH);
```

---

## SIOC\_ENABLE\_PATH

This IOCTL command enables a manually disabled (fenced) path to a device that is disabled by **SIOC\_DISABLE\_PATH** IOCTL. The `arg` parameter on the IOCTL command specifies the path to be enabled. The primary path is path 1, the first alternate path 2, the second alternate path 3, and so on. This command can be used concurrently when the device is already open by another process by using the `openx()` extended parameter `SC_TMCP`.

The **SIOC\_DEVICE\_PATHS** IOCTL command can be used to determine the paths that are enabled or manually disabled.

---

## SIOC\_SET\_PATH

This IOCTL command explicitly sets the current path to a device that the device driver uses. The `arg` parameter on the IOCTL command specifies the path to be set to the current path. The primary path is path 1, the first alternate path 2, the second alternate path 3, and so on. This command can be used concurrently when the device is already open by another process by using the `openx()` extended parameter `SC_TMCP`.

The **SIOC\_DEVICE\_PATHS** IOCTL command can be used to determine the current path the device driver is using for the device.

---

## SIOC\_DEVICE\_PATHS

This IOCTL command returns a `device_paths` structure. The number of paths are configured to a device and a `device_path_t` path structure for each configured path. The device, HBA, and path information for the primary path are configured along with all alternate SCSI paths. This IOCTL command must be used instead of the **SIOC\_QUERY\_PATH** IOCTL that is obsolete. This command can be used concurrently when the device is already open by another process by using the `openx()` extended parameter `SC_TMCP`.

The data structures are

```
struct device_path_t {
 char name[15]; /* logical device name */
 char parent[15]; /* logical parent name */
 uchar id_valid; /* obsolete and not set */
 uchar id; /* SCSI target address of device */
 uchar lun; /* SCSI logical unit of device */
 uchar bus; /* SCSI bus for device */
 uchar fcp_id_valid; /* FCP scsi/lun id fields valid */
 unsigned long long fcp_scsi_id; /* FCP SCSI id of device */
 unsigned long long fcp_lun_id; /* FCP logical unit of device */
 unsigned long long fcp_ww_name; /* FCP world wide name */
 uchar enabled; /* path enabled */
 uchar drive_port_valid; /* drive port field valid */
 uchar drive_port; /* drive port number */
 uchar fenced; /* path fenced by disable ioctl */
 uchar current_path; /* Current path assignment */
 uchar dynamic_tracking; /* FCP Dynamic tracking enabled */
 unsigned long long fcp_node_name; /* FCP node name */
 char type[16]; /* Device type and model */
 char serial[16]; /* Device serial number */
 uchar sas_id_valid; /* FCP scsi/lun id fields valid */
 char cpname[15]; /* logical name of control path drive */
 uchar last_path; /* Last failure path */
 char reserved[4];
};

struct device_paths {
 int number_paths; /* number of paths configured */
 struct device_path_t path[MAX_SCSI_PATH];
};
```

The arg parameter for the IOCTL is the address of a device\_paths structure.

The current\_path in the return structures is set to the current path the device uses for the device. If this IOCTL is issued to a medium changer smc logical driver, the cpname has the logical rmt name that is the control path drive for each smc logical path.

---

## SIOC\_QUERY\_PATH

This IOCTL command returns information about the device and SCSI paths, such as logical parent, SCSI IDs, and status of the SCSI paths.

Note: This IOCTL is obsolete but still supported. The **SIOC\_DEVICE\_PATHS** IOCTL must be used instead.

The data structure is

```
struct scsi_path {
 char primary_name[15]; /* Primary logical device name */
 char primary_parent[15]; /* Primary SCSI parent name */
 uchar primary_id; /* Primary target address of device */
 uchar primary_lun; /* Primary logical unit of device */
 uchar primary_bus; /* Primary SCSI bus for device */
 unsigned long long primary_fcp_scsi_id; /* Primary FCP SCSI id of device */
 unsigned long long primary_fcp_lun_id; /* Primary FCP logical unit of device */
 unsigned long long primary_fcp_ww_name; /* Primary FCP world wide name */
 uchar primary_enabled; /* Primary path enabled */
 uchar primary_id_valid; /* Primary id/lun/bus fields valid */
 uchar primary_fcp_id_valid; /* Primary FCP scsi/lun id fields valid */
 uchar alternate_configured; /* Alternate path configured */
 char alternate_name[15]; /* Alternate logical device name */
 char alternate_parent[15]; /* Alternate SCSI parent name */
 uchar alternate_id; /* Alternate target address of device */
 uchar alternate_lun; /* Alternate logical unit of device */
 uchar alternate_bus; /* Alternate SCSI bus for device */
 unsigned long long alternate_fcp_scsi_id; /* Alternate FCP SCSI id of device */
 unsigned long long alternate_fcp_lun_id; /* Alternate FCP logical unit of device */
 unsigned long long alternate_fcp_ww_name; /* Alternate FCP world wide name */
 uchar alternate_enabled; /* Alternate path enabled */
 uchar alternate_id_valid; /* Alternate id/lun/bus fields valid */
 uchar alternate_fcp_id_valid; /* Alternate FCP scsi/lun id fields valid */
 uchar primary_drive_port_valid; /* Primary drive port field valid */
 uchar primary_drive_port; /* Primary drive port number */
 uchar alternate_drive_port_valid; /* Alternate drive port field valid */
 uchar alternate_drive_port; /* Alternate drive port number */
 uchar primary_fenced; /* Primary fenced by disable ioctl */
 uchar alternate_fenced; /* Alternate fenced by disable ioctl */
 uchar current_path; /* Current path assignment */
 uchar primary_sas_id_valid; /* Primary FCP scsi/lun id fields valid */
 uchar alternate_sas_id_valid; /* Alternate FCP scsi/lun id fields valid */
 char reserved[55];
};
```

An example of the **SIOC\_QUERY\_PATH** command is

```

#include <sys/Atape.h>

int sioc_query_path(void)
{
 struct scsi_path path;

 printf("Querying SCSI paths...\n");

 if (ioctl(fd, SIOC_QUERY_PATH, &path) == 0)
 show_path(&path);

 return errno;
}

void show_path(struct scsi_path *path)
{
 printf("\n");
 if (path->alternate_configured)
 printf("Primary Path Information:\n");
 printf(" Logical Device..... %s\n",path->primary_name);
 printf(" SCSI Parent..... %s\n",path->primary_parent);
 if (path->primary_fcp_id_valid)
 {
 if (path->primary_id_valid)
 {
 printf(" Target ID..... %d\n",path->primary_id);
 printf(" Logical Unit..... %d\n",path->primary_lun);
 printf(" SCSI Bus..... %d\n",path->primary_bus);
 }
 printf(" FCP SCSI ID..... 0x%llx\n",path->primary_fcp_scsi_id);
 printf(" FCP Logical Unit..... 0x%llx\n",path->primary_fcp_lun_id);
 printf(" FCP World Wide Name..... 0x%llx\n",path->primary_fcp_ww_name);
 }
 else
 {
 printf(" Target ID..... %d\n",path->primary_id);
 printf(" Logical Unit..... %d\n",path->primary_lun);
 }
 if (path->primary_drive_port_valid)
 printf(" Drive Port Number..... %d\n",path->primary_drive_port);
 if (path->primary_enabled)
 printf(" Path Enabled..... Yes\n");
 else
 printf(" Path Enabled..... No \n");
 if (path->primary_fenced)
 printf(" Path Manually Disabled..... Yes\n");
 else
 printf(" Path Manually Disabled..... No \n");

 if (!path->alternate_configured)
 printf(" Alternate Path Configured..... No\n");
 else
 {
 printf(" Alternate Path Configured..... Yes\n");
 printf("\nAlternate Path Information:\n");
 printf(" Logical Device..... %s\n",path->alternate_name);
 printf(" SCSI Parent..... %s\n",path->alternate_parent);
 if (path->alternate_fcp_id_valid)
 {
 if (path->alternate_id_valid)
 {
 printf(" Target ID..... %d\n",path->alternate_id);
 printf(" Logical Unit..... %d\n",path->alternate_lun);
 printf(" SCSI Bus..... %d\n",path->alternate_bus);
 }
 printf(" FCP SCSI ID..... 0x%llx\n",path->alternate_fcp_scsi_id);
 printf(" FCP Logical Unit..... 0x%llx\n",path->alternate_fcp_lun_id);
 printf(" FCP World Wide Name..... 0x%llx\n",path->alternate_fcp_ww_name);
 }
 else
 {
 printf(" Target ID..... %d\n",path->alternate_id);
 printf(" Logical Unit..... %d\n",path->alternate_lun);
 }
 if (path->alternate_drive_port_valid)
 printf(" Drive Port Number..... %d\n",path->alternate_drive_port);
 if (path->alternate_enabled)
 printf(" Path Enabled..... Yes\n");
 else
 printf(" Path Enabled..... No \n");
 if (path->alternate_fenced)
 printf(" Path Manually Disabled..... Yes\n");
 else
 printf(" Path Manually Disabled..... No \n");
 }
}

```

---

## SIOC\_RESET\_PATH and SIOC\_CHECK\_PATH

Both of these IOCTL commands check all SCSI paths to a device that are not manually disabled by the **SIOC\_DISABLE\_PATH** IOCTL. It is done by issuing a **SCSI Inquiry** command on each path to verify communication. If the command succeeds, then the path is enabled. If it fails, the path is disabled and is not used by the device driver. This command can be used concurrently when the device is already open by another process by using the `openx()` extended parameter `SC_TMCP`.

This IOCTL command returns the same data structure as the **SIOC\_QUERY\_PATH** IOCTL command with the updated path information for the primary and first alternate path. See the **SIOC\_QUERY\_PATH** IOCTL command for a description of the data structure and output information. If more than one alternate path is configured for the device, then the **SIOC\_DEVICE\_PATHS** IOCTL must be used to determine the paths that are enabled.

An example of the **SIOC\_RESET\_PATH** command is

```
#include <sys/Atape.h>

int sioc_reset_path(void)
{
 struct scsi_path path;

 printf("Resetting SCSI paths...\n");

 if (ioctl(fd, SIOC_RESET_PATH, &path) == 0)
 show_path(&path);

 return errno;
}
```

---

## SIOC\_RESET\_DEVICE

This IOCTL command issues a SCSI target reset to the device if parallel SCSI is attached or a SCSI lun reset if FCP/SAS is attached to the device. This IOCTL command can be used to clear a SCSI Reservation that is active on the device. This command can be used concurrently when the device is already open by another process by using the `openx()` extended parameter `SC_TMCP`.

There is no argument for this IOCTL and the `arg` parameter is ignored.

---

## SIOC\_DRIVER\_INFO

This command returns the information about the currently installed Atape driver.

The following data structure is filled out and returned by the driver.

```
struct driver_info {
 uchar dd_name[16]; /* Atape driver name (Atape) */
 uchar dd_version[16]; /* Atape driver version e.g. 12.0.8.0 */
 uchar os[16]; /* Operating System (AIX) */
 uchar os_version[32]; /* Running OS Version e.g. 6.1 */
 uchar sys_arch[16]; /* Sys Architecture (POWER or others) */
 uchar reserved[32]; /* Reserved for IBM Development Use */
};
```

An example of the **SIOC\_DRIVER\_INFO** command is

```
#include <sys/Atape.h>

int sioc_driver_info()
{
 struct driver_info dd_info;

 printf("Issuing driver info...\n");

 if (!ioctl (fd, SIOC_DRIVER_INFO, &dd_info))
 {
 printf("Driver Name: %s\n", dd_info.dd_name);
 printf("Driver Version: %s\n", dd_info.dd_version);
 printf("Operating System: %s\n", dd_info.os);
 printf("OS Version: %s\n", dd_info.os_version);
 printf("System Arch: %s\n", dd_info.sys_arch);
 }
 return errno;
}
```

---

## Tape IOCTL operations

The device driver supports the tape IOCTL commands available with the base AIX® operating system. In addition, it supports a set of expanded tape IOCTL commands that give applications access to extra features and functions of the tape drives.

- [Overview](#)

---

## Overview

The following IOCTL commands are supported.



STIOCHGP  
Set the block size.

STIOCTOP  
Complete the IOCTL tape operation.

STIOCQRYP  
Query the tape device, device driver, and media parameters.

STIOCSETP  
Change the tape device, device driver, and media parameters.

STIOCSYNC  
Synchronize the tape buffers with the tape.

STIOCDM  
Display the message on the display panel.

STIOCQRYPOS  
Query the tape position and the buffered data.

STIOCSETPOS  
Set the tape position.

STIOCQRYSENSE  
Query the sense data from the tape device.

STIOCQRYINQUIRY  
Return the inquiry data.

STIOC\_LOG\_SENSE  
Return the log sense data.

STIOC\_RECOVER\_BUFFER  
Recover the buffered data from the tape device.

STIOC\_LOCATE  
Locate to the tape position.

STIOC\_READ\_POSITION  
Read the current tape position.

STIOC\_SET\_VOLID  
Set the volume name for the current mounted tape. The name is used for tape volume logging only.

STIOC\_DUMP  
Force and read a dump from the device.

STIOC\_FORCE\_DUMP  
Force a dump on the device.

STIOC\_READ\_DUMP  
Read a dump from the device.

STIOC\_LOAD\_UCODE  
Download the microcode to the device.

STIOC\_RESET\_DRIVE  
Issue a **SCSI Send Diagnostic** command to reset the tape drive.

STIOC\_FMR\_TAPE  
Create an FMR tape.

MTDEVICE  
Obtain the device number of a drive in an IBM® Enterprise Tape Library 3494.

STIOC\_PREVENT\_MEDIUM\_REMOVAL  
Prevent medium removal by an operator.

STIOC\_ALLOW\_MEDIUM\_REMOVAL  
Allow medium removal by an operator.

STIOC\_REPORT\_DENSITY\_SUPPORT  
Return supported densities from the tape device.

STIOC\_GET\_DENSITY  
Get the current write density settings from the tape device.

STIOC\_SET\_DENSITY  
Set the write density settings on the tape device.

STIOC\_CANCEL\_ERASE  
Cancel an **erase immediate** command that is in progress.

GET\_ENCRYPTION\_STATE  
This IOCTL can be used for application, system, and library-managed encryption. It allows a query only of the encryption status.

SET\_ENCRYPTION\_STATE  
This IOCTL can be used only for application-managed encryption. It sets encryption state for application-managed encryption.

SET\_DATA\_KEY  
This IOCTL can be used only for application-managed encryption. It sets the data key for application-managed encryption.

READ\_TAPE\_POSITION  
Read current tape position in either short, long, or extended form.

SET\_TAPE\_POSITION  
Set the current tape position to either a logical object or logical file position.

CREATE\_PARTITION  
Create one or more tape partitions and format the media.

QUERY\_PARTITION  
Query tape partitioning information and current active partition.

SET\_ACTIVE\_PARTITION  
Set the current active tape partition.

ALLOW\_DATA\_OVERWRITE  
Set the drive to allow a subsequent data overwrite type command at the current position or allow a **CREATE\_PARTITION** IOCTL when data safe (append-only) mode is enabled.

QUERY\_LOGICAL\_BLOCK\_PROTECTION  
Query Logical Block Protection (LBP) support and its setup.

SET\_LOGICAL\_BLOCK\_PROTECTION  
Enable or disable Logical Block Protection (LBP), set the protection method, and how the protection information is transferred.

STIOC\_READ\_ATTRIBUTE

Read attribute values from medium auxiliary memory.

STIOC\_WRITE\_ATTRIBUTE  
Write attribute values to medium auxiliary memory.

VERIFY\_TAPE\_DATA  
Read the data from tape and verify its correction.

QUERY\_RAO\_INFO  
Query the maximum number and size of User Data Segments (UDS).

GENERATE\_RAO  
Send a GRAO list to request the drive to generate a Recommended Access Order list.

RECEIVE\_RAO  
Receive a Recommended Access Order list of UDS from the drive.

QUERY\_ARCHIVE\_MODE\_UNTHREAD  
Query for Archive Mode Unthread support in tape cartridge loading.

SET\_ARCHIVE\_MODE\_UNTHREAD  
Set Archive Mode Unthread enable or disable for tape cartridge loading.

TAPE\_LOAD\_UNLOAD  
Load or unload tape cartridge with various behaviors.

GET\_VHF\_DEVICE\_STATUS  
Report very high frequency data for the tape drive and medium statuses.

These IOCTL commands and their associated structures are defined in the /usr/include/sys/Atape.h header file, which is included in the corresponding C program that uses the functions.

- [STIOCHGP](#)
- [STIOCTOP](#)
- [STIOCQRYP or STIOCSETP](#)
- [STIOCSYNC](#)
- [STIOCDM](#)
- [STIOCQRYPOS or STIOCSETPOS](#)
- [STIOCQRYSENSE](#)
- [STIOCQRYINQUIRY](#)
- [STIOC LOG SENSE](#)
- [STIOC RECOVER BUFFER](#)
- [STIOC LOCATE](#)
- [STIOC READ POSITION](#)
- [STIOC SET VALID](#)
- [STIOC DUMP](#)
- [STIOC FORCE DUMP](#)
- [STIOC READ DUMP](#)
- [STIOC LOAD UCODE](#)
- [STIOC RESET DRIVE](#)
- [STIOC FMR TAPE](#)
- [MTDEVICE \(Obtain device number\)](#)
- [STIOC PREVENT MEDIUM REMOVAL](#)
- [STIOC ALLOW MEDIUM REMOVAL](#)
- [STIOC REPORT DENSITY SUPPORT](#)
- [STIOC GET DENSITY and STIOC SET DENSITY](#)
- [STIOC CANCEL ERASE](#)
- [GET ENCRYPTION STATE](#)
- [SET ENCRYPTION STATE](#)
- [SET DATA KEY](#)
- [READ TAPE POSITION](#)
- [SET TAPE POSITION](#)
- [SET ACTIVE PARTITION](#)
- [QUERY PARTITION](#)
- [CREATE PARTITION](#)
- [ALLOW DATA OVERWRITE](#)
- [QUERY LOGICAL BLOCK PROTECTION](#)
- [SET LOGICAL BLOCK PROTECTION](#)
- [STIOC READ ATTRIBUTE](#)
- [STIOC WRITE ATTRIBUTE](#)
- [VERIFY TAPE DATA](#)
- [QUERY RAO INFO](#)
- [GENERATE RAO](#)
- [RECEIVE RAO](#)
- [QUERY ARCHIVE\\_MODE\\_UNTHREAD](#)
- [SET ARCHIVE\\_MODE\\_UNTHREAD](#)
- [TAPE\\_LOAD\\_UNLOAD](#)
- [GET VHF\\_DEVICE\\_STATUS](#)

---

## STIOCHGP

This IOCTL command sets the current block size. A block size of zero is a variable block. Any other value is a fixed block.

An example of the **STIOCHGP** command is

```
#include <sys/Atape.h>
```

```

struct stchgp stchgp;

stchgp.st_blksize = 512;

if (ioctl(tapefd, STIOCHGP, &stchgp) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

```

## STIOCTOP

This IOCTL command runs basic tape operations. The `st_count` variable is used for many of its operations. Normal error recovery applies to these operations. The device driver can issue several tries to complete them.

For all space operations, the tape position finishes on the end-of-tape side of the record or filemark for forward movement and on the beginning-of-tape side of the record or filemark for backward movement. The only exception occurs for forward and backward space record operations over a filemark if the device is configured for the AIX® record space mode.

The input data structure is

```

struct stop
{
 short st_op; /* operations defined below */
 daddr_t st_count; /* how many of them to do (if applicable) */
};

```

The `st_op` variable is set to one of the following operations.

### STOFFL

Unload the tape. The `st_count` parameter does not apply.

### STREW

Rewind the tape. The `st_count` parameter does not apply.

### STERASE

Erase the entire tape. The `st_count` parameter does not apply.

### STERASE\_IMM

Erase the entire tape with the immediate bit set. The `st_count` parameter does not apply.

This action issues the **erase** command to the device with the immediate bit set in the SCSI CDB. When this command is used, another process can cancel the erase operation by issuing the **STIOC\_CANCEL\_ERASE** IOCTL. The application that issued the **STERASE\_IMM** still waits for the **erase** command to complete like the **STERASE** `st_op` if the **STIOC\_CANCEL\_ERASE** IOCTL is not issued. Refer to for a description of the **STIOC\_CANCEL\_ERASE** IOCTL.

### STERASEGAP

Erase the gap that was written to the tape. The `st_count` parameter does not apply.

### STRETEN

Start the rewind operation. The tape devices run the retension operation automatically when needed.

### STWEOF

Write the `st_count` number of filemarks.

### STWEOF\_IMM

Write the `st_count` number of filemarks with the immediate bit set.

This action issues a **write filemark** command to the device with the immediate bit set in the SCSI CDB. The device returns immediate status and the IOCTL also returns immediately. Unlike the **STWEOF** `st_op`, any buffered write data are not flushed to tape before the filemarks are written. This action can improve the time that it takes for a **write filemark** command to complete.

### STFSF

Space forward the `st_count` number of filemarks.

### STRSF

Space backward the `st_count` number of filemarks.

### STFSR

Space forward the `st_count` number of records.

### STRSR

Space backward the `st_count` number of records.

### STTUR

Issue the **Test Unit Ready** command. The `st_count` parameter does not apply.

### STLOAD

Issue the **SCSI Load** command. The `st_count` parameter does not apply. The operation of the **SCSI Load** command varies depending on the type of device. See the appropriate hardware reference manual.

### STSEOD

Space forward to the end of the data. The `st_count` parameter does not apply. This operation is supported except on the IBM® 3490E tape devices.

### STFSSF

Space forward to the first `st_count` number of contiguous filemarks.

### STRSSF

Space backward to the first `st_count` number of contiguous filemarks.

### STEJECT

Unload the tape. The `st_count` parameter does not apply.

### STINSRT

Issue the **SCSI Load** command. The `st_count` parameter does not apply.

Note: If zero is used for operations that require the count parameter, the command is not issued to the device, and the device driver returns a successful completion.

An example of the **STIOCTOP** command is

```
#include <sys/Atape.h>
```

```

struct stop stop;

stop.st_op=STWEOF;

stop.st_count=3;

if (ioctl(tapefd, STIOCTOP, &stop) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

```

## STIOCQRYP or STIOCSETP

The **STIOCQRYP** IOCTL command allows the program to query the tape device, device driver, and media parameters. The **STIOCSETP** IOCTL command allows the program to change the tape device, device driver, and media parameters. Before the **STIOCSETP** IOCTL command is issued, use the **STIOCQRYP** IOCTL command to query and fill the fields of the data structure that you do not want to change. Then, issue the **STIOCSETP** command to change the selected fields.

Changing certain fields (such as buffered\_mode) impacts performance. If the buffered\_mode field is false, then each record that is written to the tape is transferred to the tape immediately. This operation guarantees that each record is on the tape, but it impacts performance.

## STIOCQRYP parameters that cannot be changed with the STIOCSETP IOCTL command

The following parameters that are returned by the **STIOCQRYP** IOCTL command cannot be changed by the **STIOCSETP** IOCTL command.

- trace  
This parameter is the current setting of the AIX® system tracing for channel 0. All Atape device driver events are traced in channel 0 with other kernel events. If set to On, device driver tracing is active.
- hkwrd  
This parameter is the trace hookword used for Atape events.
- write\_protect  
If the currently mounted tape is write-protected, this field is set to TRUE. Otherwise, it is set to FALSE.
- min\_blksize  
This parameter is the minimum block size for the device. The driver sets this field by issuing the **SCSI Read Block Limits** command.
- max\_blksize  
This parameter is the maximum block size for the device. The driver sets this field by issuing the **SCSI Read Block Limits** command.
- max\_scsi\_xfer  
This parameter is the maximum transfer size of the parent SCSI adapter for the device.
- acf\_mode  
If the tape device has the ACF installed, this parameter returns the current mode of the ACF. Otherwise, the value of ACF\_NONE is returned. The ACF mode can be set from the operator panel on the tape device.
- alt\_pathing  
This parameter is the configuration setting for path failover support. If the path failover support is enabled, this parameter is set to TRUE.
- medium\_type  
This parameter is the media type of the current loaded tape. Some tape devices support multiple media types and report different values in this field. See the documentation for the specific tape device to determine the possible values.
- density\_code  
This parameter is the density setting for the current loaded tape. Some tape devices support multiple densities and report the current setting in this field. See the documentation for the specific tape device to determine the possible values.
- reserve\_type  
This parameter is the configuration setting for the reservation type that the device driver uses when the device is reserved, either a **SCSI Reserve 6** command or a **SCSI Persistent Reserve** command.
- reserve\_key  
This parameter is the reservation key the device driver uses with SCSI Persistent Reserve. If a configuration reservation key was specified, then this key can be either a 1-8 ASCII character key or a 1-16 hexadecimal key. If a configuration key was not specified, then the reservation key is a 16 hexadecimal key that the device driver generates.

## Parameters that can be changed with STIOCSETP IOCTL command

The following parameters can be changed with the **STIOCSETP** IOCTL command.

- blksize  
This parameter specifies the effective block size for the tape device.
- autoloading  
This parameter turns the autoloading feature On and Off in the device driver. If set to On, the cartridge loader is treated as a large virtual tape.
- buffered\_mode  
This parameter turns the buffered mode write On and Off.
- compression

This parameter turns the hardware compression On and Off.

- 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 EOM returns the ENOSPC error code. This write operation does not complete successfully. All subsequent write operations are allowed to continue despite the check conditions that result from EOM. When the end of the physical volume is reached, EIO is returned. This parameter can be used before EOM or after EOM is reached.

- rewind\_immediate

This parameter turns the immediate bit On and Off in rewind commands. If set to On, the STREW tape operation runs faster. However, the next command takes a long time to finish unless the rewind operation is physically complete.

- logging

This parameter turns the volume logging On and Off. If set to On, the volume log data is collected and saved in the tape log file when the **Rewind** and **Unload** command is issued to the tape drive.

- valid

This parameter is the volume ID of the current loaded tape. If it is not set, the device driver initializes the valid to UNKNOWN. If logging is active, the parameter is used to identify the volume in the tape log file entry. It is reset to UNKNOWN when the tape is unloaded.

- emulate\_autoloader

This parameter turns the emulate autoloader feature On and Off.

- 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.

- logical\_write\_protect

This parameter sets or resets the logical write protect of the current tape.

Note: The tape position must be at the beginning of the tape to change this parameter from its current value.

- capacity\_scaling and capacity\_scaling\_value

The capacity\_scaling parameter queries the capacity or logical length of the current tape or on a set operation changes the current tape capacity. On a query operation, this parameter returns the current capacity for the tape. It is one of the defined values such as SCALE\_100, SCALE\_75, SCALE\_VALUE. If the query returns SCALE\_VALUE, then the capacity\_scaling\_value parameter is the current capacity. Otherwise, the capacity\_scaling parameter is the current capacity.

On a set operation, if the capacity\_scaling parameter is set to SCALE\_VALUE then the capacity\_scaling\_value parameter is used to set the tape capacity. Otherwise, one of the other defined values for the capacity\_scaling parameter is used.

Note:

1. The tape position must be at the beginning of the tape to change this parameter from its current value.
2. Changing this parameter destroys any existing data on the tape.

- retain\_reservation

When this parameter is set to 1, the device driver does not release the device reservation when the device is closed for the current open and any subsequent opens and closes until the **STIOCSETP** IOCTL is issued with retain\_reservation parameter set to 0. The device driver still reserves the device on open to make sure that the previous reservation is still valid.

- data\_safe\_mode

This parameter queries the current drive setting for data safe (append-only) mode. Also, on a set operation it changes the current data safe mode setting on the drive. On a set operation, a parameter value of zero sets the drive to normal (non-data safe) mode and a value of 1 sets the drive to data safe mode.

- disable\_sim\_logging

This parameter turns the automatic logging of tape SIM/MIM data On and Off. By default, the device driver reads Log Sense Page X'31' automatically when device sense data indicates that data is available. The data is saved in the AIX error log. Reading Log Sense Page X'31' clears the current SIM/MIM data.

Setting this bit disables the device driver from reading the Log Sense Page so an application can read and manage its own SIM/MIM data. The SIM/MIM data is saved in the AIX error log if an application reads the data with the **SIOC\_LOG\_SENSE\_PAGE** or **STIOC\_LOG\_SENSE** IOCTLs.

- read\_sili\_bit

This parameter turns the Suppress Incorrect Length Indication (SILI) bit On and Off for variable length read commands. The device driver sets this bit when the device is configured, if it detects that the adapter can support this setting. When this bit is Off, variable length read commands results in a SCSI check condition if less data is read than the read system call requested. This action can have a significant impact on read performance.

The input or output data structure is

```
struct stchgp_s
{
 int blksize; /* new block size */
 boolean trace; /* TRUE=trace on */
 uint hkwr; /* trace hook word */
 int sync_count; /* obsolete - not used */
 boolean autoloader; /* on/off autoloader feature */
 boolean buffered_mode; /* on/off buffered mode */
 boolean compression; /* on/off compression */
 boolean trailer_labels; /* on/off allow writing after EOM */
 boolean rewind_immediate; /* on/off immediate rewinds */
 boolean bus_domination; /* obsolete - not used */
 boolean logging; /* volume logging */
 boolean write_protect; /* write protected media */
 uint min_blksize; /* minimum block size */
 uint max_blksize; /* maximum block size */
 uint max_scsi_xfer; /* maximum scsi transfer len */
 char valid[16]; /* volume id */
 uchar acf_mode; /* automatic cartridge facility mode */
 #define ACF_NONE 0
 #define ACF_MANUAL 1
 #define ACF_SYSTEM 2
 #define ACF_AUTOMATIC 3
 #define ACF_ACCUMULATE 4
}
```

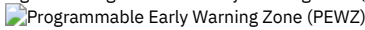
```

#define ACF_RANDOM 5
uchar record_space_mode; /* fsr/bsr space mode */
#define SCSI_SPACE_MODE 1
#define AIX_SPACE_MODE 2
uchar logical_write_protect; /* logical write protect */
#define NO_PROTECT 0
#define ASSOCIATED_PROTECT 1
#define PERSISTENT_PROTECT 2
#define WORM_PROTECT 3
uchar capacity_scaling; /* capacity scaling */
#define SCALE_100 0
#define SCALE_75 1
#define SCALE_50 2
#define SCALE_25 3
#define SCALE_VALUE 4 /* use capacity_scaling_value below */
uchar retain_reservation; /* retain reservation */
uchar alt_pathing; /* alternate pathing active */
boolean emulate_autoloader; /* emulate autoloader in random mode */
uchar medium_type; /* tape medium type */
uchar density_code; /* tape density code */
boolean disable_sim_logging; /* disable sim/mim error logging */
boolean read_sili_bit; /* SILI bit setting for read commands */
uchar capacity_scaling_value; /* capacity scaling provided value */
uchar reserve_type; /* reservation type */
#define RESERVE6_RESERVE 0 /* SCSI Reserve 6 type */
#define PERSISTENT_RESERVE 1 /* persistent reservation type */
uchar reserve_key[8]; /* persistent reservation key */
uchar data_safe_mode; /* data safe mode */
ushort pew_size; /* programmable early warning size */
uchar reserved[9];
};

```

- **pew\_size**

With the tape parameter, the application is allowed to request the tape drive to create a zone that is called the programmable early warning zone (PEWZ) in the front of Early Warning (EW).

Figure 1. Programmable Early Warning Zone (PEWZ)  


When a WRITE or WRITE FILE MARK (WFM) command writes data or filemark upon first reaching the PEWZ, Atape driver sets ENOSPC for Write and WFM to indicate that the current position reaches the PEWZ. After PEWZ is reached and before Early Warning is reached, all further writes and WFMs are allowed. The TRAILER parameter and the current design for LEOM (Logical End of Medium/Partition, or Early Warning Zone) and PEOM (Physical End of Medium/Partition) have no effect on the driver behavior in PEWZ.

For the application developers:

1. Two methods are used to determine PEWZ when the *errno* is set to ENOSPC for **Write** or **Write FileMark** command, since ENOSPC is returned for either EW or PEW.
  - Method 1: Issue a **Request Sense** IOCTL, check the sense key and ASC-ASCQ, and if it is 0x0/0x0007 (PROGRAMMABLE EARLY WARNING DETECTED), the tape is in PEW. If the sense key ASC-ASCQ is 0x0/0x0000 or 0x0/0x0002, the tape is in EW.
  - Method 2: Call **Read Position** IOCTL in long or extended form and check bpew and eop bits. If bpew = 1 and eop = 0, the tape is in PEW. If bpew = 1 and eop = 1, the tape is in EW.

Atape driver requests the tape drive to save the mode page indefinitely. The PEW size is modified in the drive until a new setup is requested from the driver or application. The application must be programmed to issue the **Set** IOCTL to zero when PEW support is no longer needed, as Atape drivers do not complete this function. PEW is a setting of the drive and not tape. Therefore, it is the same on each partition, should partitions exist.

2. Encountering the PEWZ does not cause the device server to run a synchronize operation or terminate the command. It means that the data or filemark is written in the cartridge when a check condition with PROGRAMMABLE EARLY WARNING DETECTED is returned. But, the Atape driver still returns the counter to less than zero (-1) for a **write** command or a failure for **Write FileMark** IOCTL call with ENOSPC error. In this way, it forces the application to use one of the methods to check PEW or EW. When the application determines ENOSPC comes from PEW, it reads the requested write data or filemark that are written into the cartridge and reach or pass the PEW point. The application can issue a **Read position** IOCTL to validate the tape position.

An example of the **STIOCQRYP** and **STIOCSETP** commands is

```

#include <sys/Atape.h>
struct stchgp_s stchgp;

/* get current parameters */
if (ioctl(tapefd, STIOCQRYP, &stchgp) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

/* set new parameters */
stchgp.rewind_immediate=1;
stchgp.trailer_labels=1;
if (ioctl(tapefd, STIOCSETP, &stchgp) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

```

## STIOCSYNC

This input/output control (IOCTL) command flushes the tape buffers to the tape immediately.

There are no arguments for this IOCTL command.

An example of the **STIOCSYNC** command is

```
if (ioctl(tapefd, STIOCSYNC, NULL) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOCDM

This IOCTL command displays and manipulates one or two messages on the message display. The message that is sent with this call does not always remain on the display. It depends on the current state of the tape device.

The input data structure is

```
#define MAXMSGLEN 8
struct stdm_s
{
 char dm_func; /* function code */
 /* function selection */
 #define DMSTATUSMSG 0x00 /* general status message */
 #define DMDVMSG 0x20 /* demount/verify message */
 #define DMMIMMED 0x40 /* mount with immediate action indicator */
 #define DMDEMIMMED 0xE0 /* demount/mount with immediate action */
 /* message control */
 #define DMMSG0 0x00 /* display message 0 */
 #define DMMSG1 0x04 /* display message 1 */
 #define DMFLASHMSG0 0x08 /* flash message 0 */
 #define DMFLASHMSG1 0x0C /* flash message 1 */
 #define DMALTERNATE 0x10 /* alternate message 0 and message 1 */
 char dm_msg0[MAXMSGLEN]; /* message 0 */
 char dm_msg1[MAXMSGLEN]; /* message 1 */
};
```

An example of the **STIOCDM** command is

```
#include <sys/Atape.h>
struct stdm_s stdm;
stdm.dm_func = DMSTATUSMSG | DMMSG0;
bcopy("SSD", stdm.dm_msg0, 8);
if (ioctl(tapefd, STIOCDM, &stdm) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOCQRYPOS or STIOCSETPOS

The **STIOCQRYPOS** IOCTL command queries the position on the tape. The **STIOCSETPOS** IOCTL command sets the position on the tape. Only the `block_type` and `curpos` fields are used during a set operation. The tape position is defined as where the next read or write operation occurs. The query function can be used independently or with the set function. Also, the set function can be used independently or with the query function.

The `block_type` field is set to `QP_LOGICAL` when a SCSI logical blockid format is wanted. During a query operation, the `curpos` field is set to a simple unsigned int.

On IBM® 3490 tape drives only, the `block_type` field can be set to `QP_PHYSICAL`. Setting this `block_type` on any other device is ignored and defaults to `QP_LOGICAL`. After a set operation, the position is at the logical block that is indicated by the `curpos` field. If the `block_type` field is set to `QP_PHYSICAL`, the `curpos` field that is returned is a vendor-specific blockid format from the tape device. When `QP_PHYSICAL` is used for a query operation, the `curpos` field is used only in a subsequent set operation with `QP_PHYSICAL`. This function completes a high speed locate operation. Whenever possible, use `QP_PHYSICAL` because it is faster. This advantage is obtained only when the set operation uses the `curpos` field from the `QP_PHYSICAL` query.

After a query operation, the `lbot` field indicates the last block of the data that was transferred physically to the tape. If the application writes 12 (0 - 11) blocks and `lbot` equals 8, then three blocks are in the tape buffer. This field is valid only if the last command was a **write** command. This field does not reflect the number of application write operations. A write operation can translate into multiple blocks. It reflects tape blocks as indicated by the block size. If an attempt is made to obtain this information and the last command is not a write command, the value of `LBOT_UNKNOWN` is returned.

The driver sets the `bot` field to `TRUE` if the tape position is at the beginning of the tape. Otherwise, it is set to `FALSE`. The driver sets the `eot` field to `TRUE` if the tape is positioned between the early warning and the physical end of the tape. Otherwise, it is set to `FALSE`.

The number of blocks and number of bytes currently in the tape device buffers is returned in the `num_blocks` and `num_bytes` fields. The `bcu` and `bycu` settings indicate whether these fields contain valid data. The block ID of the next block of data that transferred to or from the physical tape is returned in the `tapepos` field.

The partition number field that is returned is the current partition of the loaded tape.

The input or output data structure is

```
typedef unsigned int blockid_t;
struct stpos_s
{
 char block_type; /* format of block ID information */
 #define QP_LOGICAL 0 /* SCSI logical block ID format */
 #define QP_PHYSICAL 1 /* 3490 only, vendor-specific block ID format */
 /* ignored for all other devices */
 boolean eot; /* position is after early warning, */
 /* before physical end of tape */
 blockid_t curpos; /* for query, current position, */
};
```

```

blockid_t lbot; /* for set, position to go to */
#define LBOT_NONE 0xFFFFFFFF /* last block written to tape */
#define LBOT_UNKNOWN 0xFFFFFFFF /* unable to determine information */
uint num_blocks; /* number of blocks in buffer */
uint num_bytes; /* number of bytes in buffer */
boolean bot; /* position is at beginning of tape */
uchar partition_number; /* current partition number on tape */
boolean bcu; /* number of blocks in buffer is unknown */
boolean bycu; /* number of bytes in buffer is unknown */
blockid_t tapepos; /* next block transferred */
uchar reserved2[48];
};

```

An example of the **STIOCQRYPOS** and **STIOCSETPOS** commands is

```

#include <sys/Atape.h>
struct stpos_s stpos;
stpos.block_type=QP_PHYSICAL;
if (ioctl(tapefd, STIOCQRYPOS, &stpos) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
oldposition=stpos.curpos;
.
.
.
stpos.curpos=oldposition;
stpos.block_type=QP_PHYSICAL;
if (ioctl(tapefd, STIOCSETPOS, &stpos) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

```

---

## STIOCQRYSENSE

This IOCTL command returns the last sense data that is collected from the tape device, or it issues a new **Request Sense** command and returns the collected data. If **LASTERROR** is requested, the sense data is valid only if the last tape operation has an error that issued a sense command to the device. If the sense data is valid, the IOCTL command completes successfully and the `len` field is set to a value greater than zero.

The `residual_count` field contains the residual count from the last operation.

The input or output data structure is

```

#define MAXSENSE 255
struct stsense_s
{
 /* input */
 char sense_type; /* fresh (new sense) or sense from last error */
 #define FRESH 1 /* initiate a new sense command */
 #define LASTERROR 2 /* return sense gathered from
 the last SCSI sense command */

 /* output */
 uchar sense[MAXSENSE]; /* actual sense data */
 int len; /* length of valid sense data returned */
 int residual_count; /* residual count from last operation */
 uchar reserved[60];
};

```

An example of the **STIOCQRYSENSE** command is

```

#include <sys/Atape.h>
struct stsense_s stsense;
stsense.sense_type=LASTERROR;
#define MEDIUM_ERROR 0x03
if (ioctl(tapefd, STIOCQRYSENSE, &stsense) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
if ((stsense.sense[2] & 0x0F) == MEDIUM_ERROR)
{
 printf("We're in trouble now!");
 exit(SENSE_KEY(&stsense.sense));
}

```

---

## STIOCQRYINQUIRY

This IOCTL command returns the inquiry data from the device. The data is divided into standard and vendor-specific portions.

The output data structure is

```

/* inquiry data info */
struct inq_data_s
{

```



```

BYTE b0;
/* macros for accessing fields of byte 1 */
#define PERIPHERAL_QUALIFIER(x) ((x->b0 & 0xE0)>>5)
#define PERIPHERAL_CONNECTED 0x00
#define PERIPHERAL_NOT_CONNECTED 0x01
#define LUN_NOT_SUPPORTED 0x03

#define PERIPHERAL_DEVICE_TYPE(x) (x->b0 & 0x1F)
#define DIRECT_ACCESS 0x00
#define SEQUENTIAL_DEVICE 0x01
#define PRINTER_DEVICE 0x02
#define PROCESSOR_DEVICE 0x03
#define CD_ROM_DEVICE 0x05
#define OPTICAL_MEMORY_DEVICE 0x07
#define MEDIUM_CHANGER_DEVICE 0x08
#define UNKNOWN 0x1F

BYTE b1;
/* macros for accessing fields of byte 2 */
#define RMB(x) ((x->b1 & 0x80)>>7) /* removable media bit */
#define FIXED 0
#define REMOVABLE 1
#define device_type_qualifier(x) (x->b1 & 0x7F) /* vendor specific */

BYTE b2;
/* macros for accessing fields of byte 3 */
#define ISO_Version(x) ((x->b2 & 0xC0)>>6)
#define ECMA_Version(x) ((x->b2 & 0x38)>>3)
#define ANSI_Version(x) ((x->b2 & 0x07)
#define NONSTANDARD 0
#define SCSI1 1
#define SCSI2 2

BYTE b3;
/* macros for accessing fields of byte 4 */
#define AENC(x) ((x->b3 & 0x80)>>7) /* asynchronous event notification */
#ifndef TRUE
#define TRUE 1
#endif
#ifndef FALSE
#define FALSE 0
#endif
#define TrmIOP(x) ((x->b3 & 0x40)>>6) /* support terminate I/O process message? */
#define Response_Data_Format(x) (x->b3 & 0x0F)
#define SCSI1INQ 0 /* SCSI-1 standard inquiry data format */
#define CCSINQ 1 /* CCS standard inquiry data format */
#define SCSI2INQ 2 /* SCSI-2 standard inquiry data format */

BYTE additional_length; /* number of bytes following this field minus 4 */
BYTE res56[2];

BYTE b7;
/* macros for accessing fields of byte 7 */
#define RelAdr(x) ((x->b7 & 0x80)>>7) /* the following fields are true or false */
#define WBus32(x) ((x->b7 & 0x40)>>6)
#define WBus16(x) ((x->b7 & 0x20)>>5)
#define Sync(x) ((x->b7 & 0x10)>>4)
#define Linked(x) ((x->b7 & 0x08)>>3)
#define CmdQue(x) ((x->b7 & 0x02)>>1)
#define SftRe(x) ((x->b7 & 0x01)

```

```

char vendor_identification[8];
char product_identification[16];
char product_revision_level[4];
};
struct st_inquiry
{
 struct inq_data_s standard;
 BYTE vendor_specific[255-sizeof(struct inq_data_s)];
};

```

An example of the **STIOCQRYINQUIRY** command is

```

struct st_inquiry inqd;
if (ioctl(tapefd, STIOCQRYINQUIRY, &inqd) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
if (ANSI_Version(((struct inq_data_s *)&(inqd.standard))) == SCSI2)
 printf("Hey! We have a SCSI-2 device\n");

```

## STIOC\_LOG\_SENSE

This IOCTL command returns the log sense data from the device. If volume logging is set to On, the log sense data is saved in the tape log file.

The output data structure is

```

struct log_sense
{
 struct log_record_header header;

```

```
char data[MAXLOGSENSE];
}
```

An example of the **STIOC\_LOG\_SENSE** command is

```
struct log_sense logdata;

if (ioctl(tapefd, STIOC_LOG_SENSE, &logdata) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOC\_RECOVER\_BUFFER

This IOCTL command recovers the buffer data from the tape device. It is typically used after an error occurs during a write operation that prevents the data in the tape device buffers from being written to tape. The **STIOCQRYPOS** command can be used before this IOCTL command to determine the number of blocks and the bytes of data that is in the device buffers.

Each **STIOC\_RECOVER\_BUFFER** IOCTL call returns one block of data from the device. This ioctl command can be issued multiple times to completely recover all the buffered data from the device.

After the IOCTL command is completed, the `ret_len` field contains the number of bytes returned in the application buffer for the block. If no blocks are in the tape device buffer, then the `ret_len` value is set to zero.

The output data structure is

```
struct buffer_data
{
 char *buffer;
 int bufsize;
 int ret_len;
};
```

An example of the **STIOC\_RECOVER\_BUFFER** command is

```
struct buffer_data bufdata;

bufdata.bufsize = 256 * 1024;
bufdata.buffer = malloc(256 * 1024);

if (ioctl(tapefd, STIOC_RECOVER_BUFFER, &bufdata) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
}
else
{
 printf("Returned bytes=%d", bufdata.ret_len);
}
```

---

## STIOC\_LOCATE

This IOCTL command causes the tape to be positioned at the specified block ID. The block ID used for the command must be obtained with the **STIOC\_READ\_POSITION** command.

An example of the **STIOC\_LOCATE** command is

```
#include <sys/Atape.h>

unsigned int current_blockid;

/* read current tape position */
if (ioctl(tapefd, STIOC_READ_POSITION, &current_blockid) < 0)
{
 printf("IOCTL failure. errno=%d\n", errno);
 exit(1);
}

/* restore current tape position */
if (ioctl(tapefd, STIOC_LOCATE, current_blockid) < 0)
{
 printf("IOCTL failure. errno=%d\n", errno);
 exit(1);
}
```

---

## STIOC\_READ\_POSITION

This IOCTL command returns the block ID of the current position of the tape. The block ID returned from this command can be used with the **STIOC\_LOCATE** command to set the position of the tape.

An example of the **STIOC\_READ\_POSITION** command is

```
#include <sys/Atape.h>

unsigned int current_blockid;

/* read current tape position */
if (ioctl(tapefd, STIOC_READ_POSITION, &current_blockid) < 0)
{
 printf("IOCTL failure. errno=%d\n", errno);
 exit(1);
}

/* restore current tape position */
if (ioctl(tapefd, STIOC_LOCATE, current_blockid) < 0)
{
 printf("IOCTL failure. errno=%d\n", errno);
 exit(1);
}
```

---

## STIOC\_SET\_VOLID

This IOCTL command sets the volume name for the currently mounted tape. The volume name is used by the device driver for tape volume logging only and is not written or stored on the tape. The volume name is reset to unknown whenever an **unload** command is issued to unload the current tape. The volume name can be queried and set by using the **STIOCQRYP** and **STIOCSETP** IOCTLs.

The argument that is used for this command is a character pointer to a buffer that contains the name of the volume to be set.

An example of the **STIOC\_SET\_VOLID** command is

```
/* set the volume id for the current tape to VOL001 */
char *volid = "VOL001";
if (ioctl(tapefd, STIOC_SET_VOLID, volid) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOC\_DUMP

This IOCTL command forces a dump on the tape device, then stores the dump to either a host-specified file or in the /var/adm/ras system directory. The device driver stores up to three dumps in this directory. The first dump that is created is named Atape.rmtx.dump1, where x is the device number, for example, rmt0. The second and third dumps are dump2 and dump3. After a third dump file is created, the next dump starts at dump1 again and overlays the previous dump1 file.

The argument that is used for this command is NULL to dump to the system directory. Or, it is a character pointer to a buffer that contains the path and file name for the dump file. The dump can also be stored on a diskette by specifying /dev/rfd0 for the name.

An example of the **STIOC\_DUMP** command is

```
/* generate drive dump and store in the system directory */
if (ioctl(tapefd, STIOC_DUMP, NULL) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

/* generate drive dump and store in file 3590.dump */
char *dump_name = "3590.dump";
if (ioctl(tapefd, STIOC_DUMP, dump_name) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOC\_FORCE\_DUMP

This IOCTL command forces a dump on the tape device. The dump can be retrieved from the device by using the **STIOC\_READ\_DUMP** IOCTL.

There are no arguments for this IOCTL command.

An example of the **STIOC\_FORCE\_DUMP** command is

```
/* generate a drive dump */
if (ioctl(tapefd, STIOC_FORCE_DUMP, NULL) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOC\_READ\_DUMP

This IOCTL command reads a dump from the tape device. Then, it stores the dump to either a host specified file or in the /var/adm/ras system directory. The device driver stores up to three dumps in this directory. The first dump that is created is named Atape.rmtx.dump1, where x is the device number, for example rmt0. The second and third dumps are dump2 and dump3. After a third dump file is created, the next dump starts at dump1 again and overlays the previous dump1 file.

Dumps are either generated internally by the tape drive or can be forced by using the **STIOC\_FORCE\_DUMP** IOCTL.

The argument that is used for this command is NULL to dump to the system directory. Or, it is a character pointer to a buffer that contains the path and file name for the dump file. The dump can also be stored on a diskette by specifying /dev/rfd0 for the name.

An example of the **STIOC\_READ\_DUMP** command is

```
/* read drive dump and store in the system directory */
if (ioctl(tapefd,STIOC_READ_DUMP,NULL)<0)
{
 printf("IOCTL failure. errno=%d",errno);
 exit(errno);
}

/* read drive dump and store in file 3590.dump */
char *dump_name = "3590.dump";
if (ioctl(tapefd,STIOC_READ_DUMP,dump_name)<0)
{
 printf("IOCTL failure. errno=%d",errno);
 exit(errno);
}
```

---

## STIOC\_LOAD\_UCODE

This IOCTL command downloads microcode to the device. The argument that is used for this command is a character pointer to a buffer that contains the path and file name of the microcode. Microcode can also be loaded from a diskette by specifying /dev/rfd0 for the name.

An example of the **STIOC\_LOAD\_UCODE** command is

```
/* download microcode from file */
char *name = "/etc/microcode/D0I4_BB5.fmrz";
if (ioctl(tapefd,STIOC_LOAD_UCODE,name)<0)
{
 printf("IOCTL failure. errno=%d",errno);
 exit(errno);
}

/* download microcode from diskette */
if (ioctl(tapefd,STIOC_LOAD_UCODE,"/dev/rfd0")<0)
{
 printf("IOCTL failure. errno=%d",errno);
 exit(errno);
}
```

---

## STIOC\_RESET\_DRIVE

This IOCTL command issues a **SCSI Send Diagnostic** command to reset the tape drive. There are no arguments for this IOCTL command.

An example of the **STIOC\_RESET\_DRIVE** command is

```
/* reset the tape drive */
if (ioctl(tapefd,STIOC_RESET_DRIVE,NULL)<0)
{
 printf("IOCTL failure. errno=%d",errno);
 exit(errno);
}
```

---

## STIOC\_FMR\_TAPE

This IOCTL command creates an FMR tape. The tape is created with the current microcode loaded in the tape device.

There are no arguments for this IOCTL command.

An example of the **STIOC\_FMR\_TAPE** command is

```
/* create fmr tape */
if (ioctl(tapefd,STIOC_FMR_TAPE,NULL)<0)
{
 printf("IOCTL failure. errno=%d",errno);
 exit(errno);
}
```

---

## MTDEVICE (Obtain device number)

This IOCTL command obtains the device number that is used for communicating with the IBM® TotalStorage™ Enterprise library 3494.

The structure of the IOCTL request is

```
int device;
if (ioctl(tapefd, MTDEVICE, &device) < 0)
{
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}
```

---

## STIOC\_PREVENT\_MEDIUM\_REMOVAL

This IOCTL command prevents an operator from removing medium from the device until the **STIOC\_ALLOW\_MEDIUM\_REMOVAL** command is issued or the device is reset.

There is no associated data structure.

An example of the **STIOC\_PREVENT\_MEDIUM\_REMOVAL** command is

```
#include <sys/Atape.h>

if (!ioctl (tapefd, STIOC_PREVENT_MEDIUM_REMOVAL, NULL))
 printf ("The STIOC_PREVENT_MEDIUM_REMOVAL ioctl succeeded\n");
else
{
 perror ("The STIOC_PREVENT_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}
```

---

## STIOC\_ALLOW\_MEDIUM\_REMOVAL

This IOCTL command allows an operator to remove medium from the device. This command is used normally after an **STIOC\_PREVENT\_MEDIUM\_REMOVAL** command to restore the device to the default state.

There is no associated data structure.

An example of the **STIOC\_ALLOW\_MEDIUM\_REMOVAL** command is

```
#include <sys/Atape.h>

if (!ioctl (tapefd, STIOC_ALLOW_MEDIUM_REMOVAL, NULL))
 printf ("The STIOC_ALLOW_MEDIUM_REMOVAL ioctl succeeded\n");
else
{
 perror ("The STIOC_ALLOW_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}
```

---

## STIOC\_REPORT\_DENSITY\_SUPPORT

This IOCTL command issues the **SCSI Report Density Support** command to the tape device and returns either all supported densities or supported densities for the currently mounted media. The media field specifies which type of report is requested. The number\_reports field is returned by the device driver and indicates how many density reports in the reports array field were returned.

The data structures that are used with this IOCTL are

```
typedef struct density_report
{
 uchar primary_density_code; /* primary density code */
 uchar secondary_density_code; /* secondary density code */
 uint wrtok :1, /* write ok, device can write this format */
 dup :1, /* zero if density only reported once */
 deflt :1, /* current density is default format */
 res_1 :5; /* reserved */
 uchar reserved[2]; /* reserved */
 uchar bits_per_mm[3]; /*bits per mm */
 uint bits_per_mm:24; /* bits per mm */
 ushort media_width; /* media width in millimeters */
 ushort tracks; /* tracks */
 uint capacity; /* capacity in megabytes */
 char assigning_org[8]; /* assigning organization in ASCII */
 char density_name[8]; /* density name in ASCII */
 char description[20]; /* description in ASCII */
};

struct report_density_support
{
 uchar media; /* report all or current media as defined above */
 ushort number_reports; /* number of density reports returned in array */
 struct density_report reports[MAX_DENSITY_REPORTS];
};
```

Examples of the **STIOC\_REPORT\_DENSITY\_SUPPORT** command are

```
#include <sys/Atape.h>

int stioc_report_density_support(void)
{
 int i;
 struct report_density_support density;

 printf("Issuing Report Density Support for ALL supported media...\n");

 density.media = ALL_MEDIA_DENSITY;

 if (ioctl(fd, STIOC_REPORT_DENSITY_SUPPORT, &density) != 0)
 return errno;
 printf("Total number of densities reported: %d\n", density.number_reports);
 for (i = 0; i < density.number_reports; i++)
 {
 printf("\n");
 printf(" Density Name.....%0.8s\n",
 density.reports[i].density_name);
 printf(" Assigning Organization..%0.8s\n",
 density.reports[i].assigning_org);
 printf(" Description.....%0.20s\n",
 density.reports[i].description);
 printf(" Primary Density Code....%02X\n",
 density.reports[i].primary_density_code);
 printf(" Secondary Density Code..%02X\n",
 density.reports[i].secondary_density_code);

 if (density.reports[i].wrtok)
 printf(" Write OK.....Yes\n");
 else
 printf(" Write OK.....No\n");

 if (density.reports[i].dup)
 printf(" Duplicate.....Yes\n");
 else
 printf(" Duplicate.....No\n");

 if (density.reports[i].deflt)
 printf(" Default.....Yes\n");
 else
 printf(" Default..... No\n");

 printf(" Bits per MM..... %d\n",
 density.reports[i].bits_per_mm);
 printf(" Media Width (millimeters)%d\n",
 density.reports[i].media_width);
 printf(" Tracks..... %d\n",
 density.reports[i].tracks);
 printf(" Capacity (megabytes)....%d\n",
 density.reports[i].capacity);

 if (opcode)
 {
 printf("Hit <enter> to continue...");
 getchar();
 }
 }

 printf("\nIssuing Report Density Support for CURRENT media...\n");

 density.media = CURRENT_MEDIA_DENSITY;

 if (ioctl(fd, STIOC_REPORT_DENSITY_SUPPORT, &density) != 0)
 return errno;

 for (i = 0; i < density.number_reports; i++)
 {
 printf("\n");
 printf(" Density Name.....%0.8s\n",
 density.reports[i].density_name);
 printf(" Assigning Organization..%0.8s\n",
 density.reports[i].assigning_org);
 printf(" Description.....%0.20s\n",
 density.reports[i].description);
 printf(" Primary Density Code....%02X\n",
 density.reports[i].primary_density_code);
 printf(" Secondary Density Code..%02X\n",
 density.reports[i].secondary_density_code);

 if (density.reports[i].wrtok)
 printf(" Write OK.....Yes\n");
 else
 printf(" Write OK.....No\n");

 if (density.reports[i].dup)
 printf(" Duplicate.....Yes\n");
 else
 printf(" Duplicate.....No\n");

 if (density.reports[i].deflt)
 printf(" Default.....Yes\n");
 else
 printf(" Default.....No\n");
 }
}
```

```

printf(" Bits per MM.....%d\n",density.reports[i].bits_per_mm);
printf(" Media Width (millimeters)%d\n",density.reports[i].media_width);
printf(" Tracks.....%d\n",density.reports[i].tracks);
printf(" Capacity (megabytes)...%d\n",density.reports[i].capacity);
}

return errno;
}

```

## STIOC\_GET\_DENSITY and STIOC\_SET\_DENSITY

The **STIOC\_GET\_DENSITY** IOCTL is used to query the current write density format settings on the tape drive. The current density code from the drive Mode Sense header, the Read/Write Control Mode page default density, and pending density are returned.

The **STIOC\_SET\_DENSITY** IOCTL is used to set a new write density format on the tape drive by using the default and pending density fields. The density code field is not used and ignored on this IOCTL. The application can specify a new write density for the current loaded tape only or as a default for all tapes. Refer to the examples.

The application must get the current density settings first before the current settings are modified. If the application specifies a new density for the current loaded tape only, then the application must issue another **Set Density** IOCTL after the current tape is unloaded and the next tape is loaded to either the default maximum density or a new density. This action ensures the tape drive uses the correct density. If the application specifies a new default density for all tapes, the setting remains in effect until changed by another set density IOCTL or the tape drive is closed by the application.

Following is the structure for the **STIOC\_GET\_DENSITY** and **STIOC\_SET\_DENSITY** IOCTLs.

```

struct density_data_t
{
 char density_code; /* mode sense header density code */
 char default_density; /* default write density */
 char pending_density; /* pending write density */
 char reserved[9];
};

```

Note:

1. The IOCTLs are only supported on tape drives that can write multiple density formats. Refer to the Hardware Reference for the specific tape drive to determine whether multiple write densities are supported. If the tape drive does not support the IOCTLs, *errno* EINVAL is returned.
2. The device driver always sets the default maximum write density for the tape drive on every open system call. Any previous **STIOC\_SET\_DENSITY** IOCTL values from the last open are not used.
3. If the tape drive detects an invalid density code or cannot complete the operation on the **STIOC\_SET\_DENSITY** IOCTL, the *errno* is returned. Then, the current drive density settings before the IOCTL are restored.
4. The **struct density\_data\_t** defined in the header file is used for both IOCTLs. The *density\_code* field is not used and ignored on the **STIOC\_SET\_DENSITY** IOCTL.

## Examples

```

struct density_data_t data;

/* open the tape drive */
/* get current density settings */
rc = ioctl(fd, STIOC_GET_DENSITY, &data);

/* set 3592 J1A density format for current loaded tape only */
data.default_density = 0x7F;
data.pending_density = 0x51;
rc = ioctl(fd, STIOC_SET_DENSITY, &data);

/* unload tape */
/* load next tape */
/* set 3592 E05 density format for current loaded tape only */
data.default_density = 0x7F;
data.pending_density = 0x52;
rc = ioctl(fd, STIOC_SET_DENSITY, &data);

/* unload tape */
/* load next tape */
/* set default maximum density for current loaded tape */
data.default_density = 0;
data.pending_density = 0;
rc = ioctl(fd, STIOC_SET_DENSITY, &data);

/* close the tape drive */
/* open the tape drive */
/* set 3592 J1A density format for current loaded tape and all subsequent tapes */
data.default_density = 0x51;
data.pending_density = 0x51;

rc = ioctl(fd, STIOC_SET_DENSITY, &data);

```

## STIOC\_CANCEL\_ERASE

The **STIOC\_CANCEL\_ERASE** IOCTL is used to cancel an erase operation currently in progress. This action happens when an application issued the **STIOCTOP** IOCTL with the *st\_op* field that specifies STERASE\_IMM. The application that issued the erase and is waiting for it to complete then returns immediately with *errno* ECANCELLED. This

IOCTL always returns 0 whether an erase immediate operation is in progress or not.

This IOCTL can be issued only when the `openx()` extended parameter `SC_TMCP` is used to open the device. It happens when the application that issued the erase still has the device currently open. There is no argument for this IOCTL and the `arg` parameter is ignored.

---

## GET\_ENCRYPTION\_STATE

This IOCTL command queries the drive's encryption method and state. The data structure that is used for this IOCTL is for all of the supported operating systems.

```
struct encryption_status {
 uchar encryption_capable; /* (1) Set this field as a boolean based on the
 capability of the drive */
 uchar encryption_method; /* (2) Set this field to one of the
 #defines METHOD_* below */
#define METHOD_NONE 0 /* Only used in GET_ENCRYPTION_STATE */
#define METHOD_LIBRARY 1 /* Only used in GET_ENCRYPTION_STATE */
#define METHOD_SYSTEM 2 /* Only used in GET_ENCRYPTION_STATE */
#define METHOD_APPLICATION 3 /* Only used in GET_ENCRYPTION_STATE */
#define METHOD_CUSTOM 4 /* Only used in GET_ENCRYPTION_STATE */
#define METHOD_UNKNOWN 5 /* Only used in GET_ENCRYPTION_STATE */
 uchar encryption_state; /* (3) Set this field to one of the
 #defines STATE_* below */
#define STATE_OFF 0 /* Used in GET/SET_ENCRYPTION_STATE */
#define STATE_ON 1 /* Used in GET/SET_ENCRYPTION_STATE */
#define STATE_NA 2 /* Only used in GET_ENCRYPTION_STATE */
 uchar[13] reserved;
};
```

An example of the **GET\_ENCRYPTION\_STATE** command is

```
int qry_encryption_state (void)
{
 int rc = 0;
 struct encryption_status encryption_status_t;

 printf("issuing query encryption status...\n");
 memset(&encryption_status_t, 0, sizeof(struct encryption_status));
 rc = ioctl(fd, GET_ENCRYPTION_STATE, &encryption_status_t);
 if(rc == 0)
 {
 if(encryption_status_t.encryption_capable)
 printf("encryption capable.....Yes\n");
 else
 printf("encryption capable.....No\n");
 switch(encryption_status_t.encryption_method)
 {
 case METHOD_NONE:
 printf("encryption method.....METHOD_NONE\n");
 break;
 case METHOD_LIBRARY:
 printf("encryption method.....METHOD_LIBRARY\n");
 break;
 case METHOD_SYSTEM:
 printf("encryption method.....METHOD_SYSTEM\n");
 break;
 case METHOD_APPLICATION:
 printf("encryption method.....METHOD_APPLICATION\n");
 break;
 case METHOD_CUSTOM:
 printf("encryption method.....METHOD_CUSTOM\n");
 break;
 case METHOD_UNKNOWN:
 printf("encryption method.....METHOD_UNKNOWN\n");
 break;

 default:
 printf("encryption method.....Error\n");
 }

 switch(encryption_status_t.encryption_state)
 {
 case STATE_OFF:
 printf("encryption state.....OFF\n");
 break;
 case STATE_ON:
 printf("encryption state.....ON\n");
 break;
 case STATE_NA:
 printf("encryption state.....NA\n");
 break;

 default:
 printf("encryption state.....Error\n");
 }
 }

 return rc;
}
```



---

## SET\_ENCRYPTION\_STATE

This IOCTL command allows set encryption state only for application-managed encryption. On unload, some of the drive settings can be reset to default. To set the encryption state, the application must issue this IOCTL after a tape is loaded and at BOP.

The data structure used for this IOCTL is the same as the one for **GET\_ENCRYPTION\_STATE**. An example of the **SET\_ENCRYPTION\_STATE** command is

```
int set_encryption_state(int option)
{
 int rc = 0;
 struct encryption_status encryption_status_t;

 printf("issuing query encryption status...\n");
 memset(&encryption_status_t, 0, sizeof(struct encryption_status));
 rc = ioctl(fd, GET_ENCRYPTION_STATE, &encryption_status_t);
 if(rc < 0) return rc;

 if(option == 0)
 encryption_status_t.encryption_state = STATE_OFF;
 else if(option == 1)
 encryption_status_t.encryption_state = STATE_ON;
 else
 {
 printf("Invalid parameter.\n");
 return -EINVAL;
 }

 printf("Issuing set encryption state.....\n");
 rc = ioctl(fd, SET_ENCRYPTION_STATE, &encryption_status_t);

 return rc;
}
```

---

## SET\_DATA\_KEY

This IOCTL command allows set the data key only for application-managed encryption. The data structure that is used for this IOCTL is for all of the supported operating systems.

```
struct data_key
{
 uchar[12] data_key_index;
 uchar data_key_index_length;
 uchar[15] reserved1;
 uchar[32] data_key;
 uchar[48] reserved2;
};
```

An example of the **SET\_DATA\_KEY** command is

```
int set_datakey(void)
{
 int rc = 0;
 struct data_key encryption_data_key_t;

 printf("Issuing set encryption data key.....\n");
 memset(&encryption_data_key_t, 0, sizeof(struct data_key));
 /* fill in your data key here, then issue the following ioctl*/
 rc = ioctl(fd, SET_DATA_KEY, &encryption_data_key_t);
 return rc;
}
```

---

## READ\_TAPE\_POSITION

The **READ\_TAPE\_POSITION** IOCTL is used to return **Read Position** command data in either the short, long, or extended form. The type of data to return is specified by setting the data\_format field to either RP\_SHORT\_FORM, RP\_LONG\_FORM, or RP\_EXTENDED\_FORM.

The data structures that are used with this IOCTL are

```
#define RP_SHORT_FORM 0x00
#define RP_LONG_FORM 0x06
#define RP_EXTENDED_FORM 0x08

struct short_data_format {
 uint bop:1, /* beginning of partition */
 eop:1, /* end of partition */
 locu:1, /* 1 means num_buffer_logical_obj field is
 unknown */
 bycu:1, /* 1 means the num_buffer_bytes field is
 unknown */
 rsvd :1,
 lolu:1, /* 1 means the first and last logical
 obj position fields are unknown */
 perr: 1, /* 1 means the position fields have
 overflowed and can not be reported */
};
```

```

 bpew :1; /* beyond programmable early warning */
 uchar active_partition; /* current active partition */
 char reserved[2];
 uint first_logical_obj_position; /* current logical object position */
 uint last_logical_obj_position; /* next logical object to be transferred
 to tape */
 uint num_buffer_logical_obj; /* number of logical objects in buffer */
 uint num_buffer_bytes; /* number of bytes in buffer */
 char reserved1;
};

struct long_data_format {
 uint bop:1, /* beginning of partition */
 eop:1, /* end of partition */
 rsvd1:2,
 mpu:1, /* 1 means the logical file id field
 in unknown */
 lonu:1, /* 1 means either the partition number
 or logical obj number field
 are unknown */
 rsvd2:1,
 bpew :1; /* beyond programmable early
 warning */
 char reserved[6];
 uchar active_partition; /* current active partition */
 ullong logical_obj_number; /* current logical object
 position */
 ullong logical_file_id; /* number of filemarks from bop
 and current logical position */
 ullong obsolete;
};

struct extended_data_format {
 uint bop:1, /* beginning of partition */
 eop:1, /* end of partition */
 locu:1, /* 1 means num_buffer_logical_obj field
 is unknown */
 bycu:1, /* 1 means the num_buffer_bytes field
 is unknown */
 rsvd :1,
 lolu:1, /* 1 means the first and last logical obj
 position fields are unknown */
 perr: 1, /* 1 means the position fields have
 overflowed and can not be reported */
 bpew :1; /* beyond programmable early warning */
 uchar active_partition; /* current active partition */
 ushort additional_length;
 uint num_buffer_logical_obj; /* number of logical objects in buffer */
 ullong first_logical_obj_position; /* current logical object position */
 ullong last_logical_obj_position; /* next logical object to be transferred
 to tape */
 ullong num_buffer_bytes; /* number of bytes in buffer */
 char reserved;
};

struct read_tape_position{
 uchar data_format; /* Specifies the return data format either short,
long or extended as defined above */
 union
 {
 struct short_data_format rp_short;
 struct long_data_format rp_long;
 struct extended_data_format rp_extended;
 char reserved[64];
 } rp_data;
};

```

Example of the **READ\_TAPE\_POSITION** IOCTL

```

#include <sys/Atape.h>

struct read_tape_position rpos;

printf("Reading tape position long form...\n");
rpos.data_format = RP_LONG_FORM;
if (ioctl (fd, READ_TAPE_POSITION, &rpos) <0)
 return errno;

if (rpos.rp_data.rp_long.bop)
 printf(" Beginning of Partition Yes\n");
else
 printf(" Beginning of Partition No\n");
if (rpos.rp_data.rp_long.eop)
 printf(" End of Partition Yes\n");
else
 printf(" End of Partition No\n");
if (rpos.rp_data.rp_long.bpew)
 printf(" Beyond Early Warning Yes\n");
else
 printf(" Beyond Early Warning No\n");
if (rpos.rp_data.rp_long.lonu)
{
 printf(" Active Partition UNKNOWN \n");
 printf(" Logical Object Number UNKNOWN \n");
}
else

```

```

 {
 printf(" Active Partition %u \n",
 rpos.rp_data.rp_long.active_partition);
 printf(" Logical Object Number %llu \n",
 rpos.rp_data.rp_long.logical_obj_number);
 }

 if (rpos.rp_data.rp_long.mpu)
 printf(" Logical File ID UNKNOWN \n");
 else
 printf(" Logical File ID %llu \n",
 rpos.rp_data.rp_long.logical_file_id);

```

---

## SET\_TAPE\_POSITION

The **SET\_TAPE\_POSITION** IOCTL is used to position the tape in the current active partition to either a logical block id or logical filemark. The `logical_id_type` field in the IOCTL structure specifies either a logical block or logical filemark.

The data structure that is used with this IOCTL is

```

#define LOGICAL_ID_BLOCK_TYPE 0x00
#define LOGICAL_ID_FILE_TYPE 0x01

struct set_tape_position{
 uchar logical_id_type; /* Block or file as defined above */
 ullong logical_id; /* logical object or logical file to position to */
 char reserved[32];
};

```

Examples of the **SET\_TAPE\_POSITION** IOCTL

```

#include <sys/Atape.h>

struct set_tape_position setpos;

/* position to logical block id 10 */
setpos.logical_id_type = LOGICAL_ID_BLOCK_TYPE;
setpos.logical_id = 10;
ioctl(fd, SET_TAPE_POSITION, &setpos);

/* position to logical filemark 4 */
setpos.logical_id_type = LOGICAL_ID_FILE_TYPE;
setpos.logical_id = 4;
ioctl(fd, SET_TAPE_POSITION, &setpos);

```

---

## SET\_ACTIVE\_PARTITION

The **SET\_ACTIVE\_PARTITION** IOCTL is used to position the tape to a specific partition. Then, it becomes the current active partition for subsequent commands and a specific logical block id in the partition. To position to the beginning of the partition, the `logical_block_id` field is set to 0.

The data structure that is used with this IOCTL is

```

struct set_active_partition {
 uchar partition_number; /* Partition number 0-n to change to */
 ullong logical_block_id; /* Blockid to locate to within partition */
 char reserved[32];
};

```

Examples of the **SET\_ACTIVE\_PARTITION** IOCTL

```

#include <sys/Atape.h>

struct set_active_partition partition;

/* position the tape to partition 1 and logical block id 12 */
partition.partition_number = 1;
partition.logical_block_id = 12;
ioctl(fd, SET_ACTIVE_PARTITION, &partition);

/* position the tape to the beginning of partition 0 */
partition.partition_number = 0;
partition.logical_block_id = 0;
ioctl(fd, SET_ACTIVE_PARTITION, &partition);

```

---

## QUERY\_PARTITION

The **QUERY\_PARTITION** IOCTL is used to return partition information for the tape drive and the current media in the tape drive. It includes the current active partition the tape drive is using for the media. The `number_of_partitions` field is the current number of partitions on the media and the `max_partitions` is the maximum partitions that the tape drive supports. The `size_unit` field can be either one of the defined values or another value such as 8. It is used with the `size_array` field value for each partition to specify the actual size partition sizes. The `partition_method` field is either Wrap-wise Partitioning or Longitudinal Partitioning. Refer to [CREATE\\_PARTITION](#) for details.

The data structure that is used with this IOCTL is

```

The define for "partition_method":
#define UNKNOWN_TYPE 0 /* vendor-specific or unknown */
#define WRAP_WISE_PARTITION 1 /* Wrap-wise Partitioning without RABF */
#define LONGITUDINAL_PARTITION 2 /* Longitudinal Partitioning */
#define WRAP_WISE_PARTITION_WITH_FASTSYNC 3 /* Wrap-wise Partitioning with RABF */

The define for "size_unit":
#define SIZE_UNIT_BYTES 0 /* Bytes */
#define SIZE_UNIT_KBYTES 3 /* Kilobytes */
#define SIZE_UNIT_MBYTES 6 /* Megabytes */
#define SIZE_UNIT_GBYTES 9 /* Gigabytes */
#define SIZE_UNIT_TBBYTES 12 /* Terabytes */

struct query_partition {
 uchar max_partitions; /* Max number of supported partitions */
 uchar active_partition; /* current active partition on tape */
 uchar number_of_partitions; /* Number of partitions from 1 to max */
 uchar size_unit; /* Size unit of partition sizes below */
 ushort size[MAX_PARTITIONS]; /* Array of partition sizes in size units */
 /* for each partition, 0 to (number - 1) */
 uchar partition_method; /* partitioning type */
 char reserved [31];
};

```

Examples of the **QUERY\_PARTITION** IOCTL

```

#include <sys/Atape.h>

struct query_partition partition;
int i;

if (ioctl(fd, QUERY_PARTITION, &partition) < 0)
 return errno;

printf(" Max supported partitions ... %d\n",partition.max_partitions);
printf(" Number of partitions %d\n",partition.number_of_partitions);
printf(" Active partition %d\n",partition.active_partition);
printf(" Partition Method %d\n",partition.partition_method);
if (partition.size_unit == SIZE_UNIT_BYTES)
 printf(" Partition size unit Bytes\n");
else if (partition.size_unit == SIZE_UNIT_KBYTES)
 printf(" Partition size unit Kilobytes\n");
else if (partition.size_unit == SIZE_UNIT_MBYTES)
 printf(" Partition size unit Megabytes\n");
else if (partition.size_unit == SIZE_UNIT_GBYTES)
 printf(" Partition size unit Gigabytes\n");
else if (partition.size_unit == SIZE_UNIT_TBBYTES)
 printf(" Partition size unit Terabytes\n");
else
 printf(" Partition size unit %d\n",partition.size_unit);

for (i=0; i < partition.number_of_partitions; i++)
 printf(" Partition %d size %d\n",i,partition.size[i]);

```

## CREATE\_PARTITION

The **CREATE\_PARTITION** IOCTL is used to format the current media in the tape drive into 1 or more partitions. The number of partitions to create is specified in the `number_of_partitions` field. When more than one partition is created, the type field specifies the type of partitioning, either FDP, SDP, or IDP. The tape must be positioned at the beginning of tape (partition 0 logical block id 0) before this IOCTL is used.

If the `number_of_partitions` field to create in the IOCTL structure is one partition, all other fields are ignored and not used. The tape drive formats the media by using its default partitioning type and size for a single partition.

When the type field in the IOCTL structure is set to either FDP or SDP, the `size_unit` and `size` fields in the IOCTL structure are not used. When the type field in the IOCTL structure is set to IDP, the `size_unit` with the `size` fields are used to specify the size for each partition.

There are two partition types: Wrap-wise Partitioning ([Figure 1](#)) optimized for streaming performance, and Longitudinal Partitioning ([Figure 2](#)) optimized for random access performance. Media is always partitioned into 1 by default. Or, more than one partition where the data partition always exists as partition 0 and other extra index partition 1 to n can exist.

A WORM media cannot be partitioned and the **Format Medium** commands are rejected. Attempts to scale a partitioned media is accepted. However, only if you use the correct **FORMAT** field setting, as part of scaling the volume is set to a single data partition cartridge.

Figure 1. Wrap-wise partitioning

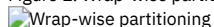
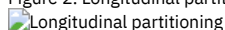


Figure 2. Longitudinal partitioning



The following chart lists the maximum number of partitions that the tape drive supports.

Table 1. Number of supported partitions

| Drive type                          | Maximum number of supported partitions |
|-------------------------------------|----------------------------------------|
| LTO 5 (TS2250 and TS2350) and later | 2 in Wrap-wise Partitioning            |
| 3592 E07 (TS 1140)                  | 4 in Wrap-wise Partitioning            |
|                                     | 2 in Longitudinal Partitioning         |

The data structure that is used with this IOCTL is

```

The define for "partition_method":
#define UNKNOWN_TYPE 0 /* vendor-specific or unknown */
#define WRAP_WISE_PARTITION 1 /* Wrap-wise Partitioning without RABF */
#define LONGITUDINAL_PARTITION 2 /* Longitudinal Partitioning */
#define WRAP_WISE_PARTITION_WITH_FASTSYNC 3 /* Wrap-wise Partitioning with RABF */

The define for "type":
#define IDP_PARTITION 1 /* Initiator Defined Partition type */
#define SDP_PARTITION 2 /* Select Data Partition type */
#define FDP_PARTITION 3 /* Fixed Data Partition type */

The define for "size unit":
#define SIZE_UNIT_BYTES 0 /* Bytes */
#define SIZE_UNIT_KBYTES 3 /* Kilobytes */
#define SIZE_UNIT_MBYTES 6 /* Megabytes */
#define SIZE_UNIT_GBYTES 9 /* Gigabytes */
#define SIZE_UNIT_TBBYTES 12 /* Terabytes */

struct tape_partition {
 uchar type; /* Type of tape partition to create */
 uchar number_of_partitions; /* Number of partitions to create */
 uchar size_unit; /* IDP size unit of partition sizes below */
 ushort size[MAX_PARTITIONS]; /* Array of partition sizes in size units */
 /* for each partition, 0 to (number - 1) */
 uchar partition_method; /* partitioning type */
 char reserved [31];
};

```

Examples of the **CREATE\_PARTITION** IOCTL

```

#include <sys/Atape.h>

struct tape_partition partition;

/* create 2 SDP partitions on LTO-5 */
partition.type = SDP_PARTITION;
partition.number_of_partitions = 2;
partition.partition_method = WRAP_WISE_PARTITION;
ioctl(fd, CREATE_PARTITION, &partition);

/* create 2 IDP partitions with partition 1 for 37 gigabytes and partition 0
for the remaining capacity on LTO-5 */
partition.type = IDP_PARTITION;
partition.number_of_partitions = 2;
partition.partition_method = WRAP_WISE_PARTITION;
partition.size_unit = SIZE_UNIT_GBYTES;
partition.size[0] = 0xFFFF;
partition.size[1] = 37;
ioctl(fd, CREATE_PARTITION, &partition);

/* format the tape into 1 partition */
partition.number_of_partitions = 1;
ioctl(fd, CREATE_PARTITION, &partition);

/* create 4 IDP partitions on 3592 JC volume in Wrap-wise partitioning
with partition 0 and 2 for 94.11 gigabytes (minimum size) and partition 1 and 3
to use the remaining capacity equally around 1.5 TB on 3592 E07 */
partition.type = IDP_PARTITION;
partition.number_of_partitions = 4;
partition.partition_method = WRAP_WISE_PARTITION;
partition.size_unit = 8; /* 100 megabytes */
partition.size[0] = 0x03AD;
partition.size[1] = 0xFFFF;
partition.size[2] = 0x03AD;
partition.size[3] = 0x3AD2;

```

## ALLOW\_DATA\_OVERWRITE

The **ALLOW\_DATA\_OVERWRITE** IOCTL is used to set the drive to allow a subsequent data write type command at the current position. Or, it allows a **CREATE\_PARTITION** IOCTL when data safe (append-only) mode is enabled.

For a subsequent write type command, the `allow_format_overwrite` field must be set to 0. The `partition_number` and `logical_block_id` fields must be set to the current partition and position within the partition where the overwrite occurs.

For a subsequent **CREATE\_PARTITION** IOCTL, the `allow_format_overwrite` field must be set to 1. The `partition_number` and `logical_block_id` fields are not used. However, the tape must be at the beginning of tape (partition 0 logical block id 0) before the **CREATE\_PARTITION** IOCTL is issued.

The data structure that is used with this IOCTL is

```

struct allow_data_overwrite{
 uchar partition_number; /* Partition number 0-n to overwrite */
 ullong logical_block_id; /* Blockid to overwrite to within partition */
 uchar allow_format_overwrite; /* allow format if in data safe mode */
 char reserved[32];
};

```

Examples of the **ALLOW\_DATA\_OVERWRITE** IOCTL

```

#include <sys/Atape.h>

struct read_tape_position rpos;

```

```

struct allow_data_overwrite data_overwrite;
struct set_active_partition partition;

/* get current tape position for a subsequent write type command and */
/* set the allow_data_overwrite fields with the current position for the next
write type command */
rpos.data_format = RP_LONG_FORM;
if (ioctl (fd, READ_TAPE_POSITION, &rpos) < 0)
 return errno;

data_overwrite.partition_number = rpos.rp_data.rp_long.active_partition;
data_overwrite.logical_block_id = rpos.rp_data.rp_long.logical_obj_number;
data_overwrite.allow_format_overwrite = 0;
ioctl (fd, ALLOW_DATA_OVERWRITE, &data_overwrite);

/* set the tape position to the beginning of tape and */
/* prepare a format overwrite for the CREATE_PARTITION ioctl */
partition.partition_number = 0;
partition.logical_block_id = 0;
if (ioctl (fd, SET_ACTIVE_PARTITION, &partition) < 0)
 return errno;

data_overwrite.allow_format_overwrite = 1;
ioctl (fd, ALLOW_DATA_OVERWRITE, &data_overwrite);

```

---

## QUERY\_LOGICAL\_BLOCK\_PROTECTION

The IOCTL queries whether the drive can support this feature, what Logical Block Protection (LBP) method is used, and where the protection information is included.

The lbp\_capable field indicates whether the drive has logical block protection capability. The lbp\_method field displays if LBP is enabled and what the protection method is. The LBP information length is shown in the lbp\_info\_length field. The fields of lbp\_w, lbp\_r, and rbdp present that the protection information is included in write, read, or recover buffer data.

The data structure that is used with this IOCTL is

```

struct logical_block_protection
{
 uchar lbp_capable; /* [OUTPUT] the capability of lbp for QUERY ioctl only */
 uchar lbp_method; /* lbp method used for QUERY [OUTPUT] and SET [INPUT] ioctls */
 #define LBP_DISABLE 0x00
 #define REED_SOLOMON_CRC 0x01
 uchar lbp_info_length; /* lbp info length for QUERY [OUTPUT] and SET [INPUT] ioctls */
 uchar lbp_w; /* protection info included in write data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 uchar lbp_r; /* protection info included in read data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 uchar rbdp; /* protection info included in recover buffer data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 uchar reserved[26];
};

```

Examples of the **QUERY\_LOGICAL\_BLOCK\_PROTECTION** IOCTL

```

#include <sys/Atape.h>

struct logical_block_protection lbp_protect;

printf("Querying Logical Block Protection...\n");

if (ioctl (fd, QUERY_LOGICAL_BLOCK_PROTECTION, &lbp_protect) < 0)
 return errno;
printf(" Logical Block Protection capable..... %d\n", lbp_protect.lbp_capable);
printf(" Logical Block Protection method..... %d\n", lbp_protect.lbp_method);
printf(" Logical Block Protection Info Length... %d\n", lbp_protect.lbp_info_length);
printf(" Logical Block Protection for Write..... %d\n", lbp_protect.lbp_w);
printf(" Logical Block Protection for Read..... %d\n", lbp_protect.lbp_r);
printf(" Logical Block Protection for RBDP..... %d\n", lbp_protect.rbdp);

```

---

## SET\_LOGICAL\_BLOCK\_PROTECTION

The IOCTL enables or disables Logical Block Protection, sets up what method is used, and where the protection information is included.

The lbp\_capable field is ignored in this IOCTL by the Atape driver. If the lbp\_method field is 0 (LBP\_DISABLE), all other fields are ignored and not used. When the lbp\_method field is set to a valid non-zero method, all other fields are used to specify the setup for LBP.

The data structure that is used with this IOCTL is

```

struct logical_block_protection
{
 uchar lbp_capable; /* [OUTPUT] the capability of lbp for QUERY ioctl only */
 uchar lbp_method; /* lbp method used for QUERY [OUTPUT] and SET [INPUT] ioctls */
 #define LBP_DISABLE 0x00
 #define REED_SOLOMON_CRC 0x01
 uchar lbp_info_length; /* lbp info length for QUERY [OUTPUT] and SET [INPUT] ioctls */
 uchar lbp_w; /* protection info included in write data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 uchar lbp_r; /* protection info included in read data */
};

```

```

 uchar rbdp; /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 /* protection info included in recover buffer data */
 uchar reserved[26]; /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
};

```

Examples of the **SET\_LOGICAL\_BLOCK\_PROTECTION** IOCTL

```

#include <sys/Atape.h>

int rc;
struct logical_block_protection lbp_protect;

printf("Setting Logical Block Protection....\n\n");

printf ("Enter Logical Block Protection method: ");
gets (buf);
lbp_protect.lbp_method= atoi(buf);
printf ("Enter Logical Block Protection Info Length: ");
gets (buf);
lbp_protect.lbp_info_length= atoi(buf);
printf ("Enter Logical Block Protection for Write: ");
gets (buf);
lbp_protect.lbp_w= atoi(buf);
printf ("Enter Logical Block Protection for Read: ");
gets (buf);
lbp_protect.lbp_r= atoi(buf);
printf ("Enter Logical Block Protection for RBDP: ");
gets (buf);
lbp_protect.rbdp= atoi(buf);

rc = ioctl(fd, SET_LOGICAL_BLOCK_PROTECTION, &lbp_protect);

if (rc)
 printf ("Set Logical Block Protection Fails (rc %d)",rc);
else
 printf ("Set Logical Block Protection Succeeds");

```

Note:

1. The drive always expects a CRC attached with a data block when LBP is enabled for lbp\_r and lbp\_w. Without the CRC bytes attachment, the drive fails the **Read and Write** command. To prevent the CRC block transfer between the drive and application, the maximum block size limit must be determined by application. Call the **STIOCQRYVP** IOCTL and get the system maximum block size limit. Call the **Read Block Limits** command to get the drive maximum block size limit. Then, use the minimum of the two limits.
2. When a unit attention with a power-on and device reset (Sense key/Asc-Ascq x6/x2900) occurs, the LBP enable bits (lbp\_w, lbp\_r, and rbdp) are reset to OFF by default. Atape tape driver returns EIO for an IOCTL call in this situation. Once the application determines it is a reset unit attention in the sense data, it responses to query LBP setup and reissues this IOCTL to set up LBP properly.
3. The LBP setting is controlled by the application and not the device driver. If an application enables LBP, it must also disable LBP when it closes the drive, as this action is not done by the device driver.

## STIOC\_READ\_ATTRIBUTE

The IOCTL is issued to read attribute values that belongs to a specific partition from medium auxiliary memory.

The input or output data structure is

```

#define MAX_ATTR_LEN 1024
struct read_attribute
{
 uchar service_action; /* [IN] service action */
 uchar partition_number; /* [IN] the partition which the attributes belong to */
 ushort first_attr_id; /* [IN] first attribute id to be returned */
 uint attr_data_len; /* [OUT] length of attribute data returned */
 uchar reserved[8];
 char data[MAX_ATTR_LEN]; /* [OUT] read attributes data */
};

```

An example of the **STIOC\_READ\_ATTRIBUTE** command is

```

#include <sys/Atape.h>
int rc,attr_len;
struct read_attribute rd_attr;

memset(&rd_attr,0,sizeof(struct read_attribute));
rd_attr.service_action=0x00;
rd_attr.partition_number=1;
rd_attr.first_attr_id=0x800;

printf("Read attribute command\n");
rc=ioctl(fd, STIOC_READ_ATTRIBUTE, &rd_attr);

if (rc)
 printf ("Read Attribute failed (rc %d)",rc);
else
{
 printf ("Read Attribute Succeeds!");
 dump_bytes (rd_attr.data, min(MAX_ATTR_LEN, rd_attr.attr_data_len),
"Attribute Data");
}

```

---

## STIOC\_WRITE\_ATTRIBUTE

The IOCTL sets the attributes in medium auxiliary memory at a specific partition.

Following is the structure for **STIOC\_WRITE\_ATTRIBUTE** IOCTL

```
struct write_attribute
{
 uchar write_cache; /* [IN] WTC - Write-through cache */
 uchar partition_number; /* [IN] the partition which the attribute is belonged to */
 uint parm_list_len; /* [IN] parameter list length */
 uchar reserved[10];
 char data[MAX_ATTR_LEN]; /* [IN] write attributes data */
};
```

An example of the **STIOC\_WRITE\_ATTRIBUTE** command is

```
#include <sys/Atape.h>

int rc;
struct write_attribute wr_attr;

memset(&wr_attr,0,sizeof(struct write_attribute));

wr_attr.write_cache=0;
wr_attr.parm_list_len=0x11;
wr_attr.data[3]=0x0D;
wr_attr.data[4]=0x08;
wr_attr.data[6]=0x01;
wr_attr.data[8]=0x08;
wr_attr.data[9]='I';
wr_attr.data[10]='B';
wr_attr.data[11]='M';
wr_attr.data[12]=' ';
wr_attr.data[13]='T';
wr_attr.data[14]='E';
wr_attr.data[15]='S';
wr_attr.data[16]='T';

printf("Issuing a sample Write Attribute command\n\n");
rc=ioctl(fd, STIOC_WRITE_ATTRIBUTE, &wr_attr);

if (rc)
 printf ("Write Attribute failed (rc %d)",rc);
else
 printf ("Write Attribute Succeeds");
```

---

## VERIFY\_TAPE\_DATA

The IOCTL issues a **VERIFY** command. This command causes data to be read from the tape and passed through the drive's error detection and correction hardware. This action determines whether it can be recovered from the tape, whether the protection information is present, and validates correctly on logical block on the medium. The driver returns the IOCTL a failure or a success if the **VERIFY SCSI** command is completed in a Good SCSI status.

Note:

1. When an application sets the VBF method, it considers the driver's close operation in which the driver can write filemarks in its close, which the application did not explicitly request. For example, some drivers write two consecutive filemarks that mark the end of data on the tape in its close, if the last tape operation was a **WRITE** command.
2. Per the user's or application's request, Atape driver sets the block size in the field of Block Length in mode block descriptor for **Read and Write** commands. Then, it maintains this block size setting in a whole open. For instance, the tape driver sets a zero in the Block Length field for the variable block size. This act causes the missing of an overlength condition on a SILI Read. Block Length must be set to a non-zero value.  
Before Fixed bit is set to ON with VTE or VBF ON in **Verify** IOCTL, the application is requested to set the block size in mode block descriptor. Then, the drive uses it to verify the length of each logical block. For example, a 256 KB length is set in Block Length field to verify the data. The setup overrides the early setting from IBM® tape driver.  
  
When the application completes the **Verify** IOCTL call, the original block size setting must be restored for **Read and Write** commands, the application either issues **set block size** IOCTL. Or, it closes the drive immediately and reopens the drive for the next tape operation. It is recommended to reopen the drive for the next tape operation. Otherwise, it causes next **Read and Write** command misbehavior.
3. To support DPF for **Verify** command with FIXED bit on, it is requested to issue an IBM tape driver to set blksize in **STIOCSETP** IOCTL. IBM tape driver sets the block length in mode block descriptor same as the block size and save the block size in kernel memory. The driver restores the block length before it tries the **Verify SCSI** command again. Otherwise, it causes the **Verify** command to fail.
4. The IOCTL can be returned longer than the timeout when DPF occurs.

The structure is defined for this IOCTL as

```
struct verify_data
{
 uint : 2, /* reserved */
 vte: 1, /* [IN] verify to end-of-data */
 vlbp: 1, /* [IN] verify logical block protection info */
 vbf: 1, /* [IN] verify by filemarks */
 immed: 1, /* [IN] return SCSI status immediately */
 bytcmp: 1, /* No use currently */
 fixed: 1; /* [IN] set Fixed bit to verify the length of each logical block */
};
```



```
uchar reseved[15];
uint verify_length; /* [IN] amount of data to be verified */
};
```

An example of the **VERIFY\_TAPE\_DATA** command is to verify all of logical block from the current position to end of data. It includes a verification that each logical block uses the logical block protection method that is specified in the Control Data Protection mode page, when vte is set to 1 with vlbpm on.

```
#include <sys/Atape.h>

int rc;
struct verify_data vrf_data;

memset(&vrf_data,0,sizeof(struct verify_data));
vrf_data.vte=1;
vrf_data.vlbpm=1;
vrf_data.vbf=0;
vrf_data.immed=0;
vrf_data.fixed=0;
vrf_data.verify_length=0;

printf("Verify Tape Data command\n");
rc=ioctl(fd,VERIFY_TAPE_DATA, &vrf_data);

if (rc)
printf ("Verify Tape Data failed (rc %d)",rc);
else printf
("Verify Tape Data Succeeded!");
```

---

## QUERY\_RAO\_INFO

The IOCTL is used to query the maximum number and size of User Data Segments (UDS) that are supported from tape drive and driver for the wanted uds\_type. The application calls this IOCTL before the **GENERATE\_RAO** and **RECEIVE\_RAO** IOCTLs are issued. The application uses the return data to limit the number of UDS requested in the **GENERATE\_RAO** IOCTL.

The structure that is defined for this IOCTL as

```
struct query_rao_info {
 char uds_type; /* [IN] 0: UDS_WITHOUT_GEOMETRY */
 /* 1: UDS_WITH_GEOMETRY */
 char reserved[7];
 ushort max_uds_number; /* [OUT] Max UDS number supported from drive */
 ushort max_uds_size; /* [OUT] Max single UDS size supported from drive in byte */
 ushort max_host_uds_number; /* [OUT] Max UDS number supported from driver */
};
```

An example of the **QUERY\_RAO\_INFO** command is

```
#include <sys/Atape.h>

int rc;
struct query_rao_info qrao;

bzero(&qrao,sizeof(struct query_rao_info));

qrao.uds_type=uds_type;

rc=ioctl(fd,QUERY_RAO_INFO,&qrao);

if (rc)
printf("QUERY_RAO_INFO fails with rc %d\n",rc);
else
{
 max_host_uds_num=qrao.max_host_uds_number;
 max_uds_size=qrao.max_uds_size;
}

return rc;
```

---

## GENERATE\_RAO

The IOCTL is called to send a GRAO list to request the drive to generate a Recommending Access Order list.

The process method is either 1 or 2 to create a RAO list, and the type of UDS is either with or without the geometry. The uds\_number must be not larger than **max\_host\_uds\_number** in the **QUERY\_RAO\_INFO** IOCTL. The application allocates a memory with **grao\_list\_leng (uds\_number \* sizeof(struct grao\_uds\_desc) +8)** for the pointer of **grao\_list**. 8 bytes is the size that is needed for the header portion on of the return data.

The structures for the **GENERATE\_RAO** IOCTL are

```
struct generate_rao {
 char process; /* [IN] Requested process to generate RAO list */
 /* 0: no reorder UDS and no calculate locate time */
 /* (not currently supported by the drive) */
 /* 1: no reorder UDS but calculate locate time */
 /* 2: reorder UDS and calculate locate time */
 char uds_type; /* [IN] 0: UDS_WITHOUT_GEOMETRY */
 /* 1: UDS_WITH_GEOMETRY */
};
```

```

char reserved1[2];
uint grao_list_leng; /* [IN] The data length is allocated for GRAO list. */
char *grao_list; /* [IN] the pointer is allocated to the size of grao_list_leng */
/* (uds_number * sizeof(struct grao_uds_desc)
/* +sizeof(struct grao_list_header))
/* and contains the data of GRAO parameter list.
/* The uds_number isn't larger than max_host_uds_number
/* in QUERY_RAO ioctl.
char reserved2[8];
}

```

The grao list is in the format and the parameter data can be generated by using the structures that are defined here.

```

-- List Header
-- UDS Segment Descriptor (first)
.....
-- UDS Segment Descriptor (last)

struct grao_list_header {
 uchar reserved[4];
 uint addl_data; /* additional data */
}

struct grao_uds_desc {
 ushort desc_leng; /* descriptor length */
 char reserved[3];
 char uds_name[10]; /* uds name given by application */
 char partition; /* Partition number 0-n to overwrite */
 ullong beginning_loi; /* Beginning logical object ID */
 ullong ending_loi; /* Ending logical object ID */
}

```

An example of the **GENERATE\_RAO** command is

```

#include<sys/Atape.h>

int rc;
struct generate_rao grao;

bzero(&grao,sizeof(struct generate_rao));

grao.process=2;
grao.uds_type=uds_type;
grao.grao_list_leng=max_host_uds_num * sizeof(struct grao_uds_desc)
+ sizeof(struct grao_list_header);

if (!(grao.grao_list=malloc(grao.grao_list_leng)))
{
 perror("Failure allocating memory");
 return (errno);
}

memset(grao.grao_list, 0, grao.grao_list_leng);
grao.grao_list[3]=36;

rc=ioctl(fd,GENERATE_RAO,&grao);
if (rc)
 printf("GENERATE_RAO fails with rc %d\n",rc);
else
 printf("GENERATE_RAO succeeds\n");

free(grao.grao_list);

return rc;

```

## RECEIVE\_RAO

After a **GENERATE\_RAO** IOCTL is completed, the application calls the **RECEIVE\_RAO** IOCTL to receive a recommended access order of UDS from the drive. To avoid a system crash, it is important that the application allocates a large enough block of memory for the *\*rrao\_list* pointer and notifies the driver of the allocated size. It is done by indicating the size of the buffer in bytes to the *rrao\_list\_leng* variable as an input to the *receive\_rao\_list* structure.

The structure for the **RECEIVE\_RAO** IOCTL is

```

struct receive_rao_list {
 uint rrao_list_offset; /* [IN] The offset of receive RAO list to */
/* begin returning data */
 uint rrao_list_leng; /* [IN/OUT] number byte of data length */
/* [IN] The data length is allocated for RRAO */
/* list by application the length is */
/* (max_uds_size * uds_number + */
/* sizeof(struct rrao_list_header) */
/* max_uds_size is reported in */
/* sizeof(struct rrao_list_header) */
/* uds_number is the total UDS number */
/* requested from application in */
/* GENERATE_RAO ioctl */
/* [OUT] the data length is actual returned */
/* in RRAO list from the driver */
 char *rrao_list; /* [IN/OUT] the data pointer of RRAO list */
 char reserved[8];
};

```

The rrao list is in this format.

```
List Header

UDS descriptor (first)
-- Basic UDS descriptor
-- Additional UDS info descriptor (first)
.....
-- Additional UDS info descriptor (last)

.....

UDS descriptor (last)
-- Basic UDS descriptor
-- Additional UDS info descriptor (first)
.....
-- Additional UDS info descriptor (last)
```

The sample code is

```
int rc;
struct receive_rao_list rrao;

bzero(&rrao, sizeof(struct receive_rao_list));

rrao.rrao_list_offset=0;
rrao.rrao_list_leng=max_host_uds_num * max_uds_size + 8;
/* 8 is the header of rrao list */

if (!(rrao.rrao_list=malloc(rrao.rrao_list_leng)))
{
 perror("Failure allocating memory");
 return (errno);
}
memset(rrao.rrao_list, 0, rrao.rrao_list_leng);

rc=ioctl(fd, RECEIVE_RAO, &rrao);
if (rc)
 printf("RECEIVE_RAO fails with rc %d\n", rc);
else
 printf("rrao_list_leng %d\n", rrao.rrao_list_leng);

free(rrao.rrao_list);

return rc;
```

---

## QUERY\_ARCHIVE\_MODE\_UNTHREAD

This ioctl reports the feature of Archive Mode Unthread enable or disable in the tape drive.

The structure for this IOCTL is

```
struct archive_mode_unthread
{
 uchar amu_enable;
 uchar reserve[7];
} archive_mode_unthread
```

An example of the **QUERY\_ARCHIVE\_MODE\_UNTHREAD** command is

```
#include <sys/Atape.h>
int rc;
struct archive_mode_unthread qamu;
bzero(&qamu, sizeof(struct archive_mode_unthread));
rc = ioctl(fd, QUERY_ARCHIVE_MODE_UNTHREAD, &qamu);
if (rc)
 printf("QUERY_ARCHIVE_MODE_UNTHREAD fails with rc %d\n", rc);
else
 printf("amu_enable %d \n", qamu.amu_enable);
return rc;
```

---

## SET\_ARCHIVE\_MODE\_UNTHREAD

This ioctl is used to enable or disable the feature of Archive Mode Unthread in the tape drive.

The structure is defined for this IOCTL as

```
struct archive_mode_unthread
{
 uchar amu_enable;
 uchar reserve[7];
} archive_mode_unthread;
```

An example of the **SET\_ARCHIVE\_MODE\_UNTHREAD** command is

```
#include <sys/Atape.h>
int rc;
```

```

struct archive_mode_unthread set_amu;
set_amu.amu_enable=ENABLE;
rc=ioctl(fd,SET_ARCHIVE_MODE_UNTHREAD,&set_amu);
if (rc)
printf("SET_ARCHIVE_MODE_UNTHREAD fails with rc %d\n",rc);
else
printf("SET_ARCHIVE_MODE_UNTHREAD succeeds \n");
return rc;

```

## TAPE\_LOAD\_UNLOAD

This ioctl is called to complete the various behaviors of tape cartridge loading. When Archive Mode Unthread is enabled, the variable of *RETEN* can be used to set the new unthread method in this new load command. See [Table 1](#) for interaction with other variables.

| Table 1. Behavior for the combinations of the retention, load, hold variables |      |       |           |                                                                                                                                                                                              |
|-------------------------------------------------------------------------------|------|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Volume position                                                               | Hold | Load* | Retention | Description                                                                                                                                                                                  |
| U, M, I, or T                                                                 | 0    | 0     | 0         | Unload the cartridge from the drive. Upon completion of the command, MAM is not accessible. If the cartridge is already unloaded, GOOD Status is returned.                                   |
| U, M, or I                                                                    | 0    | 0     | 1         | Unload the cartridge from the drive. Upon completion of the command, MAM is not accessible. If the cartridge is already unloaded, GOOD Status is returned.                                   |
| T                                                                             | 0    | 0     | 1         | Run an archive mode Unthread (see <a href="#">SET_ARCHIVE_MODE_UNTHREAD</a> ) and then unload the cartridge from the drive. Upon completion of the command, MAM is not accessible.           |
| U, M, or I                                                                    | 0    | 1     | -         | Load the cartridge and become READY.                                                                                                                                                         |
| T                                                                             | 0    | 1     | -         | The logical position is set to logical block 0 of partition 0 (BOP 0) (Not equivalent to a <b>Rewind</b> command as the active partition is set to partition 0).                             |
| U, M, I, or T                                                                 | 1    | -     | 1         | The cartridge is moved to the seated position with MAM accessible and the tape not threaded.                                                                                                 |
| U, M, or I                                                                    | 1    | -     | 1         | The cartridge is moved to the seated position with MAM accessible and the tape not threaded.                                                                                                 |
| T                                                                             | 1    | -     | 1         | An archive mode Unthread (see <a href="#">SET_ARCHIVE_MODE_UNTHREAD</a> ) is completed and then the cartridge is moved to the seated position with MAM accessible and the tape not threaded. |

Key:  
U - Unloaded  
M - MAM accessible not threaded  
I = In the IDLE\_C power condition state (that is, power-saving mode) with a volume mounted  
T - Threaded  
- = Don't care

\*The **LOAD UNLOAD** command with the LOAD bit set to 0b is sometimes called an **unload** command. The **LOAD UNLOAD** command with the LOAD bit set to 1b is sometimes called a **reload** command.

The structure for the **TAPE\_LOAD\_UNLOAD** ioctl is defined as:

```

struct tape_load_unload
{
uchar immediate; /* [IN] immediate return a SCSI status */
uchar hold; /* [IN] request MAM accessible and volume not for access */
uchar eot; /* [IN] end of tape, no use and always 0 now */
uchar retention; /* [IN] archive mode unthread */
uchar load; /* [IN] perform a load or unload */
char reserved[11];
};

```

An example of the **TAPE\_LOAD\_UNLOAD** command is

```

#include <sys/Atape.h>
int rc;
struct tape_load_unload load_unload;
printf("Tape Load Unload Command....\n");
printf ("Enter immediate : ");
gets (buf);
load_unload.immediate = atoi(buf);
printf ("Enter hold : ");
gets (buf);
load_unload.hold = atoi(buf);
printf ("Enter retention : ");
gets (buf);
load_unload.retention = atoi(buf);
printf ("Enter load : ");
gets (buf);
load_unload.load = atoi(buf);
printf ("\n");
rc = ioctl(fd, TAPE_LOAD_UNLOAD, &load_unload);
if (rc)
printf ("Tape Load Unload Command Fails (rc %d)",rc);
else
printf ("Tape Load Unload Command Succeeds");
return rc;

```

## GET\_VHF\_DEVICE\_STATUS

The device drivers provide application a friendly interface **GET\_VHF\_DEVICE\_STATUS** to report the tape drive and medium statuses. (VHF is short Very High Frequency).

The structure for the **GET\_VHF\_DEVICE\_STATUS** ioctl is

```
struct get_vhf_device_status
{
 uchar pamr; /* [OUT] prevent/allow medium removal */
 uchar hui; /* [OUT] host initialed unload */
 uchar macc; /* [OUT] medium auxiliary memory (MAM) accessible */
 uchar cmpr; /* [OUT] data compress is enabled */
 uchar wrtp; /* [OUT] volume is physically write protected when mprcnt is on */
 uchar crqst; /* [OUT] cleaning requested */
 uchar crqrd; /* [OUT] cleaning required */
 uchar dinit; /* [OUT] this VHF data valid or invalid */
 uchar inxtn; /* [OUT] device is transitioning to another state */
 uchar rsvd1;
 uchar raa; /* [OUT] robotic access allowed */
 uchar mprcnt; /* [OUT] medium present */
 uchar rsvd2;
 uchar mstd; /* [OUT] medium seated */
 uchar mthrd; /* [OUT] medium threaded */
 uchar mounted; /* [OUT] medium is mounted */
 uchar dev_act; /* [OUT] the current activity of the device below */
 /* 00h No DT device activity
 01h Cleaning operation in progress
 02h Medium is being loaded
 03h Medium is being unloaded
 04h Other medium activity
 05h Reading from medium
 06h Writing to medium
 07h Locating medium
 08h Rewinding medium
 09h Erasing medium
 0Ah Formatting medium
 0Bh Calibrating medium
 0Ch Other DT device activity
 0Dh Microcode update in progress
 0Eh Reading encrypted from medium
 0Fh Writing encrypted to medium */
 uchar vs;
 uchar rsvd3;
 uchar tddec; /* [OUT] tape diagnostic data entry created */
 uchar epp; /* [OUT] encryption parameters present */
 uchar esr; /* [OUT] encryption service request */
 uchar rrqst; /* [OUT] recovery requested */
 uchar intfc; /* [OUT] interface changed */
 uchar tafc; /* [OUT] tape alert state flag changed */
 /* the fields below for LTO only */
 uchar rsvd4[6];
 uchar overwrite; /* [OUT] overwrite occurs and write mode isn't in append-only
mode */
 uchar pcl_p; /* [OUT] potential conflict list present */
 uchar rsvd5[6];
 uchar hostlogin; /* [OUT] host login has occurred */
 uchar sleepmode; /* [OUT] operating in one of the low power states */
 char reserved[15];
};
```

Note: IBM tape drivers reports the data from the tape drive. The device drivers still return no error on this ioctl when the tape drive returns a good SCSI status even though VHF data is invalid, so the application must determine if the data is valid before to use it. The variable of dinit indicates this VHF data valid or invalid. The variable of wrtp shows a volume is physically write protected only when mprcnt is ON.

An example of the **GET\_VHF\_DEVICE\_STATUS** command is

```
#include <sys/Atape.h>
int rc;
struct get_vhf_device_status qvhf;
bzero(&qvhf, sizeof(struct get_vhf_device_status));
rc = ioctl(fd, GET_VHF_DEVICE_STATUS, &qvhf);
if (rc)
 printf("GET_VHF_DEVICE_STATUS fails with rc %d\n", rc);
else
 printf ("GET_VHF_DEVICE_STATUS Command Succeeds");
return rc;
```

---

## Medium changer IOCTL operations

This chapter describes the set of IOCTL commands that provides control and access to the SCSI medium changer functions. These IOCTL operations can be issued to the tape special file (such as rmt0), through a separate special file (such as rmt0.smc) that was created during the configuration process, or a separate special file (such as smc0), to access the medium changer.

When an application opens a /dev/rmt special file that is assigned to a drive that has access to a medium changer, these IOCTL operations are also available. The interface to the /dev/rmt\*.smc special file provides the application access to a separate medium changer device. When this special file is open, the medium changer is treated as a separate device. While /dev/rmt\*.smc is open, access to the IOCTL operations is restricted to /dev/rmt\*.smc and any attempt to access them through /dev/rmt\* fails.

- [Overview](#)

---

## Overview

The following IOCTL commands are supported.

**SMCIOE\_ELEMENT\_INFO**  
Obtain the device element information.

**SMCIOE\_MOVE\_MEDIUM**  
Move a cartridge from one element to another element.

**SMCIOE\_EXCHANGE\_MEDIUM**  
Exchange a cartridge in an element with another cartridge.

**SMCIOE\_POS\_TO\_ELEM**  
Move the robot to an element.

**SMCIOE\_INIT\_ELEM\_STAT**  
Issue the **SCSI Initialize Element Status** command.

**SMCIOE\_INIT\_ELEM\_STAT\_RANGE**  
Issue the **SCSI Initialize Element Status with Range** command.

**SMCIOE\_INVENTORY**  
Return the information about the four element types.

**SMCIOE\_LOAD\_MEDIUM**  
Load a cartridge from a slot into the drive.

**SMCIOE\_UNLOAD\_MEDIUM**  
Unload a cartridge from the drive and return it to a slot.

**SMCIOE\_PREVENT\_MEDIUM\_REMOVAL**  
Prevent medium removal by the operator.

**SMCIOE\_ALLOW\_MEDIUM\_REMOVAL**  
Allow medium removal by the operator.

**SMCIOE\_READ\_ELEMENT\_DEVIDS**  
Return the device ID element descriptors for drive elements.

**SMCIOE\_READ\_CARTIDGE\_LOCATION**  
Returns the cartridge location information for storage elements in the library.

These IOCTL commands and their associated structures are defined by including the `/usr/include/sys/Atape.h` header file in the C program by using the functions.

- [SMCIOE\\_ELEMENT\\_INFO](#)
- [SMCIOE\\_MOVE\\_MEDIUM](#)
- [SMCIOE\\_EXCHANGE\\_MEDIUM](#)
- [SMCIOE\\_POS\\_TO\\_ELEM](#)
- [SMCIOE\\_INIT\\_ELEM\\_STAT](#)
- [SMCIOE\\_INIT\\_ELEM\\_STAT\\_RANGE](#)
- [SMCIOE\\_INVENTORY](#)
- [SMCIOE\\_LOAD\\_MEDIUM](#)
- [SMCIOE\\_UNLOAD\\_MEDIUM](#)
- [SMCIOE\\_PREVENT\\_MEDIUM\\_REMOVAL](#)
- [SMCIOE\\_ALLOW\\_MEDIUM\\_REMOVAL](#)
- [SMCIOE\\_READ\\_ELEMENT\\_DEVIDS](#)
- [SMCIOE\\_READ\\_CARTIDGE\\_LOCATION](#)

---

## SMCIOE\_ELEMENT\_INFO

This IOCTL command obtains the device element information.

The data structure is

```
struct element_info
{
 ushort robot_addr; /* first robot address */
 ushort robots; /* number of medium transport elements */
 ushort slot_addr; /* first medium storage element address */
 ushort slots; /* number of medium storage elements */
 ushort ie_addr; /* first import/export element address */
 ushort ie_stations; /* number of import/export elements */
 ushort drive_addr; /* first data-transfer element address */
 ushort drives; /* number of data-transfer elements */
};
```

An example of the **SMCIOE\_ELEMENT\_INFO** command is

```
#include <sys/Atape.h>

struct element_info element_info;

if (!ioctl (smcfd, SMCIOE_ELEMENT_INFO, &element_info))
{
 printf ("The SMCIOE_ELEMENT_INFO ioctl succeeded\n");
 printf ("\nThe element information data is:\n");
 dump_bytes ((uchar *)&element_info, sizeof (struct element_info));
}
else
{
 perror ("The SMCIOE_ELEMENT_INFO ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_MOVE\_MEDIUM

This IOCTL command moves a cartridge from one element to another element.

The data structure is

```
struct move_medium
{
 ushort robot; /* robot address */
 ushort source; /* move from location */
 ushort destination; /* move to location */
 char invert; /* invert before placement bit */
};
```

An example of the **SMCIOC\_MOVE\_MEDIUM** command is

```
#include <sys/Atape.h>

struct move_medium move_medium;

move_medium.robot = 0;
move_medium.invert = 0;
move_medium.source = source;
move_medium.destination = dest;

if (!ioctl (smcfd, SMCIOC_MOVE_MEDIUM, &move_medium))
 printf ("The SMCIOC_MOVE_MEDIUM ioctl succeeded\n");
else
{
 perror ("The SMCIOC_MOVE_MEDIUM ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_EXCHANGE\_MEDIUM

This IOCTL command exchanges a cartridge in an element with another cartridge. This command is equivalent to two **SCSI Move Medium** commands. The first moves the cartridge from the source element to the destination1 element. The second moves the cartridge that was previously in the destination1 element to the destination2 element. The destination2 element can be the same as the source element.

The input data structure is

```
struct exchange_medium
{
 ushort robot; /* robot address */
 ushort source; /* source address for exchange */
 ushort destination1; /* first destination address for exchange */
 ushort destination2; /* second destination address for exchange */
 char invert1; /* invert before placement into destination1 */
 char invert2; /* invert before placement into destination2 */
};
```

An example of the **SMCIOC\_EXCHANGE\_MEDIUM** command is

```
#include <sys/Atape.h>

struct exchange_medium exchange_medium;

exchange_medium.robot = 0;
exchange_medium.invert1 = 0;
exchange_medium.invert2 = 0;
exchange_medium.source = 32; /* slot 32 */
exchange_medium.destination1 = 16; /* drive address 16 */
exchange_medium.destination2 = 35; /* slot 35 */

/* exchange cartridge in drive address 16 with cartridge from slot 32 and */
/* return the cartridge currently in the drive to slot 35 */
if (!ioctl (smcfd, SMCIOC_EXCHANGE_MEDIUM, &exchange_medium))
 printf ("The SMCIOC_EXCHANGE_MEDIUM ioctl succeeded\n");
else
{
 perror ("The SMCIOC_EXCHANGE_MEDIUM ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_POS\_TO\_ELEM

This IOCTL command moves the robot to an element.

The input data structure is

```
struct pos_to_elem
{
 ushort robot; /* robot address */
 ushort destination; /* move to location */
};
```

```

 char invert; /* invert before placement bit */
};

```

An example of the **SMCIOC\_POS\_TO\_ELEM** command is

```

#include <sys/Atape.h>

char buf[10];
struct pos_to_elem pos_to_elem;

pos_to_elem.robot = 0;
pos_to_elem.invert = 0;
pos_to_elem.destination = dest;

if (!ioctl (smcfd, SMCIOC_POS_TO_ELEM, &pos_to_elem))
 printf ("The SMCIOC_POS_TO_ELEM ioctl succeeded\n");
else
{
 perror ("The SMCIOC_POS_TO_ELEM ioctl failed");
 smcioc_request_sense();
}

```

---

## SMCIOC\_INIT\_ELEM\_STAT

This IOCTL command instructs the medium changer robotic device to issue the **SCSI Initialize Element Status** command.

There is no associated data structure.

An example of the **SMCIOC\_INIT\_ELEM\_STAT** command is

```

#include <sys/Atape.h>

if (!ioctl (smcfd, SMCIOC_INIT_ELEM_STAT, NULL))
 printf ("The SMCIOC_INIT_ELEM_STAT ioctl succeeded\n");
else
{
 perror ("The SMCIOC_INIT_ELEM_STAT ioctl failed");
 smcioc_request_sense();
}

```

---

## SMCIOC\_INIT\_ELEM\_STAT\_RANGE

This IOCTL command issues the **SCSI Initialize Element Status with Range** command. It is used to audit specific elements in a library by specifying the starting element address and number of elements. Use the **SMCIOC\_INIT\_ELEM\_STAT** IOCTL to audit all elements.

The data structure is

```

struct element_range
{
 ushort element_address; /* starting element address */
 ushort number_elements; /* number of elements */
}

```

An example of the **SMCIOC\_INIT\_ELEM\_STAT\_RANGE** command is

```

#include <sys/Atape.h>

struct element_range elements;

/* audit slots 32 to 36 */
elements.element_address = 32;
elements.number_elements = 5;

if (!ioctl (smcfd, SMCIOC_INIT_ELEM_STAT_RANGE, &elements))
 printf ("The SMCIOC_INIT_ELEM_STAT_RANGE ioctl succeeded\n");
else
{
 perror ("The SMCIOC_INIT_ELEM_STAT_RANGE ioctl failed");
 smcioc_request_sense();
}

```

---

## SMCIOC\_INVENTORY

This IOCTL command returns information about the four element types. The software application processes the input data (the number of elements about which it requires information). Then, it allocates a buffer large enough to hold the output for each element type.

The input data structure is

```

struct element_status
{
 ushort address; /* element address */
 uint :2, /* reserved */
 inenab:1, /* media into changer's scope */
}

```



```

 exenab:1, /* media out of changer's scope */
 access:1, /* robot access allowed */
 except:1, /* abnormal element state */
 impexp:1, /* import/export placed by operator or robot */
 full:1, /* element contains medium */
 uchar resvd1; /* reserved */
 uchar asc; /* additional sense code */
 uchar ascq; /* additional sense code qualifier */
 uint notbus:1, /* element not on same bus as robot */
 :1, /* reserved */
 idvalid:1, /* element address valid */
 luvalid:1, /* logical unit valid */
 :1, /* reserved */
 lun:3; /* logical unit number */
 uchar scsi; /* SCSI bus address */
 uchar resvd2; /* reserved */
 uint svalid:1, /* element address valid */
 invert:1, /* medium inverted */
 :6; /* reserved */
 ushort source; /* source storage element address */
 uchar volume[36]; /* primary volume tag */
 uchar resvd3[4]; /* reserved */
};

struct inventory
{
 struct element_status *robot_status; /* medium transport element pages */
 struct element_status *slot_status; /* medium storage element pages */
 struct element_status *ie_status; /* import/export element pages */
 struct element_status *drive_status; /* data-transfer element pages */
};

```

An example of the **SMCIOCI\_INVENTORY** command is

```

#include <sys/Atape.h>

ushort i;
struct element_status robot_status[1];
struct element_status slot_status[20];
struct element_status ie_status[1];
struct element_status drive_status[1];
struct inventory inventory;

bzero((caddr_t)robot_status,sizeof(struct element_status));

for (i=0;i<20;i++)
 bzero((caddr_t)(&slot_status[i]),sizeof(struct element_status));

bzero((caddr_t)ie_status,sizeof(struct element_status));
bzero((caddr_t)drive_status,sizeof(struct element_status));

smcioc_element_info();

inventory.robot_status = robot_status;
inventory.slot_status = slot_status;
inventory.ie_status = ie_status;
inventory.drive_status = drive_status;

if (!ioctl (smcfd, SMCIOC_INVENTORY, &inventory))
{
 printf ("\nThe SMCIOC_INVENTORY ioctl succeeded\n");
 printf ("\nThe robot status pages are:\n");

 for (i = 0; i < element_info.robots; i++)
 {
 dump_bytes ((uchar *) (inventory.robot_status+i),
 sizeof (struct element_status));
 printf ("\n--- more ---");
 getchar();
 }

 printf ("\nThe slot status pages are:\n");

 for (i = 0; i < element_info.slots; i++)
 {
 dump_bytes ((uchar *) (inventory.slot_status+i),
 sizeof (struct element_status));
 printf ("\n--- more ---");
 getchar();
 }

 printf ("\nThe ie status pages are:\n");

 for (i = 0; i < element_info.ie_stations; i++)
 {
 dump_bytes ((uchar *) (inventory.ie_status+i),
 sizeof (struct element_status));
 printf ("\n--- more ---");
 getchar();
 }

 printf ("\nThe drive status pages are:\n");

 for (i = 0; i < element_info.drives; i++)
 {
 dump_bytes ((uchar *) (inventory.drive_status+i),

```

```

 sizeof (struct element_status));
 printf ("\n--- more ---");
 getchar();
}
}
else
{
 perror ("The SMCIIOC_INVENTORY ioctl failed");
 smcioc_request_sense();
}
}

```

---

## SMCIIOC\_LOAD\_MEDIUM

This IOCTL command loads a tape from a specific slot into the drive. Or, it loads from the first full slot into the drive if the slot address is specified as zero.

An example of the **SMCIIOC\_LOAD\_MEDIUM** command is

```

#include <sys/Atape.h>

/* load cartridge from slot 3 */
if (ioctl (tapefd, SMCIIOC_LOAD_MEDIUM,3)<0)
{
 printf ("IOCTL failure. errno=%d\n",errno)
 exit(1);
}

/* load first cartridge from magazine */
if (ioctl (tapefd, SMCIIOC_LOAD_MEDIUM,0)<0)
{
 printf ("IOCTL failure. errno=%d\n",errno)
 exit(1);
}

```

---

## SMCIIOC\_UNLOAD\_MEDIUM

This IOCTL command moves a tape from the drive and returns it to a specific slot. Or, it moves a tape to the first empty slot in the magazine if the slot address is specified as zero. If the IOCTL is issued to the /dev/rmt special file, the tape is automatically rewound and unloaded from the drive first.

An example of the **SMCIIOC\_UNLOAD\_MEDIUM** command is

```

#include <sys/Atape.h>

/* unload cartridge to slot 3 */
if (ioctl (tapefd, SMCIIOC_UNLOAD_MEDIUM,3)<0)
{
 printf ("IOCTL failure. errno=%d\n",errno)
 exit(1);
}

/* unload cartridge to first empty slot in magazine */
if (ioctl (tapefd, SMCIIOC_UNLOAD_MEDIUM,0)<0)
{
 printf ("IOCTL failure. errno=%d\n",errno)
 exit(1);
}

```

---

## SMCIIOC\_PREVENT\_MEDIUM\_REMOVAL

This IOCTL command prevents an operator from removing medium from the device until the **SMCIIOC\_ALLOW\_MEDIUM\_REMOVAL** command is issued or the device is reset.

There is no associated data structure.

An example of the **SMCIIOC\_PREVENT\_MEDIUM\_REMOVAL** command is

```

#include <sys/Atape.h>

if (!ioctl (smcfd, SMCIIOC_PREVENT_MEDIUM_REMOVAL, NULL))
 printf ("The SMCIIOC_PREVENT_MEDIUM_REMOVAL ioctl succeeded\n");
else
{
 perror ("The SMCIIOC_PREVENT_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}

```

---

## SMCIIOC\_ALLOW\_MEDIUM\_REMOVAL

This IOCTL command allows an operator to remove medium from the device. This command is used normally after an **SMCIIOC\_PREVENT\_MEDIUM\_REMOVAL** command to restore the device to the default state.

There is no associated data structure.

An example of the **SMCIOCL\_ALLOW\_MEDIUM\_REMOVAL** command is

```
#include <sys/Atape.h>

if (!ioctl (smcfd, SMCIOCL_ALLOW_MEDIUM_REMOVAL, NULL))
 printf ("The SMCIOCL_ALLOW_MEDIUM_REMOVAL ioctl succeeded\n");
else
{
 perror ("The SMCIOCL_ALLOW_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOCL\_READ\_ELEMENT\_DEVIDS

This IOCTL command issues the **SCSI Read Element Status** command with the device ID (DVCID) bit set and returns the element descriptors for the data transfer elements. The `element_address` field specifies the starting address of the first data transfer element. The `number_elements` field specifies the number of elements to return. The application must allocate a return buffer large enough for the `number_elements` specified in the input structure.

The input data structure is

```
struct read_element_devids
{
 ushort element_address; /* starting element address */
 ushort number_elements; /* number of elements */
 struct element_devid *drive_devid; /* data transfer element pages */
};
```

The output data structure is

```
struct element_devid
{
 ushort address; /* element address */
 uint :4, /* reserved */
 access:1, /* robot access allowed */
 except:1, /* abnormal element state */
 :1, /* reserved */
 full:1; /* element contains medium */
 uchar resvd1; /* reserved */
 uchar asc; /* additional sense code */
 uchar ascq; /* additional sense code qualifier */
 uint notbus:1, /* element not on same bus as robot */
 :1, /* reserved */
 idvalid:1, /* element address valid */
 luvalid:1, /* logical unit valid */
 :1, /* reserved */
 lun:3; /* logical unit number */
 uchar scsi; /* scsi bus address */
 uchar resvd2; /* reserved */
 uint svalid:1, /* element address valid */
 invert:1, /* medium inverted */
 :6; /* reserved */
 ushort source; /* source storage element address */
 uint :4, /* reserved */
 code_set:4; /* code set X'2' is all ASCII identifier */
 uint :4, /* reserved */
 ident_type:4; /* identifier type */
 uchar resvd3; /* reserved */
 uchar ident_len; /* identifier length */
 uchar identifier[36]; /* device identification */
};
```

An example of the **SMCIOCL\_READ\_ELEMENT\_DEVIDS** command is

```
#include <sys/Atape.h>

int smcioc_read_element_devids()
{
 int i;
 struct element_devid *elem_devid, *elem;
 struct read_element_devids devides;
 struct element_info element_info;

 if (ioctl(fd, SMCIOCL_ELEMENT_INFO, &element_info))
 return errno;

 if (element_info.drives)
 {
 elem_devid = malloc(element_info.drives * sizeof(struct element_devid));
 if (elem_devid == NULL)
 {
 errno = ENOMEM;
 return errno;
 }
 bzero((caddr_t)elem_devid, element_info.drives * sizeof(struct element_devid));
 devides.drive_devid = elem_devid;
 devides.element_address = element_info.drive_addr;
 devides.number_elements = element_info.drives;

 printf("Reading element device ids...\n");
 }
```

```

if (ioctl (fd, SMCIOC_READ_ELEMENT_DEVIDS, &devids))
{
 free(elem_devid);
 return errno;
}

elem = elem_devid;
for (i = 0; i < element_info.drives; i++, elem++)
{
 printf("\nDrive Address %d\n",elem->address);
 if (elem->except)
 printf(" Drive State Abnormal\n");
 else
 printf(" Drive State Normal\n");
 if (elem->asc == 0x81 && elem->ascq == 0x00)
 printf(" ASC/ASCQ %02X%02X (Drive Present)\n",
 elem->asc,elem->ascq);
 else if (elem->asc == 0x82 && elem->ascq == 0x00)
 printf(" ASC/ASCQ %02X%02X (Drive Not Present)\n",
 elem->asc,elem->ascq);
 else
 printf(" ASC/ASCQ %02X%02X\n",
 elem->asc,elem->ascq);
 if (elem->full)
 printf(" Media Present Yes\n");
 else
 printf(" Media Present No\n");
 if (elem->access)
 printf(" Robot Access Allowed Yes\n");
 else
 printf(" Robot Access Allowed No\n");
 if (elem->svalid)
 printf(" Source Element Address %d\n",elem->source);
 else
 printf(" Source Element Address Valid ... No\n");
 if (elem->invert)
 printf(" Media Inverted Yes\n");
 else
 printf(" Media Inverted No\n");
 if (elem->notbus)
 printf(" Same Bus as Medium Changer No\n");
 else
 printf(" Same Bus as Medium Changer Yes\n");
 if (elem->idvalid)
 printf(" SCSI Bus Address %d\n",elem->scsi);
 else
 printf(" SCSI Bus Address Valid No\n");
 if (elem->luvalid)
 printf(" Logical Unit Number %d\n",elem->lun);
 else
 printf(" Logical Unit Number Valid No\n");
 printf(" Device ID %0.36s\n", elem->identifier);
}
}
else
{
 printf("\nNo drives found in element information\n");
}

free(elem_devid);
return errno;
}

```

## SMCIOC\_READ\_CARTIDGE\_LOCATION

The **SMCIOC\_READ\_CARTIDGE\_LOCATION** IOCTL is used to return the cartridge location information for storage elements in the library. The `element_address` field specifies the starting element address to return. The `number_elements` field specifies how many storage elements are returned. The data field is a pointer to the buffer for return data. The buffer must be large enough for the number of elements that are returned. If the storage element contains a cartridge, then the ASCII identifier field in return data specifies the location of the cartridge.

Note: This IOCTL is supported only on the TS3500 (3584) library.

The data structures that are used with this IOCTL are

```

struct cartridge_location_data
{
 ushort address; /* element address */
 uint :4, /* reserved */
 access:1, /* robot access allowed */
 except:1, /* abnormal element state */
 :1, /* reserved */
 full:1; /* element contains medium */
 uchar resvd1; /* reserved */
 uchar asc; /* additional sense code */
 uchar ascq; /* additional sense code qualifier */
 uchar resvd2[3]; /* reserved */
 uint svalid:1; /* element address valid */
 uint invert:1; /* medium inverted */
 :6; /* reserved */
 ushort source; /* source storage element address */
 uchar volume[36]; /* primary volume tag */
 uint :4, /* reserved */

```

```

 code_set:4; /* code set X'2' is all ASCII identifier */
 uint :4, /* reserved */
 ident_type:4; /* identifier type */
 uchar resvd3; /* reserved */
 uchar ident_len; /* identifier length */
 uchar identifier[24]; /* slot identification */
};

struct read_cartridge_location
{
 ushort element_address; /* starting element address */
 ushort number_elements; /* number of elements */
 struct cartridge_location_data *data; /* storage element pages */
 char reserved[8]; /* reserved */
};

```

Example of the **SMCIOC\_READ\_CARTRIDGE\_LOCATION** IOCTL

```

#include <sys/Atape.h>

int i;
struct cartridge_location_data *data, *elem;
struct read_cartridge_location cart_location;
struct element_info element_info;

/* get the number of slots and starting element address */
if (ioctl(fd, SMCIOC_ELEMENT_INFO, &element_info) < 0)
 return errno;

if (element_info.slots == 0)
 return 0;

data = malloc(element_info.slots * sizeof(struct cartridge_location_data));
if (data == NULL)
 return ENOMEM;

/* Read cartridge location for all slots */
bzero(data, element_info.slots * sizeof(struct cartridge_location_data));
cart_location.data = data;
cart_location.element_address = element_info.slot_addr;
cart_location.number_elements = element_info.slots;

if (ioctl (fd, SMCIOC_READ_CARTRIDGE_LOCATION, &cart_location) < 0)
{
 free(data);
 return errno;
}

elem = data;
for (i = 0; i < element_info.slots; i++, elem++)
{
 if (elem->address == 0)
 continue;

 printf("Slot Address %d\n", elem->address);
 if (elem->except)
 printf(" Slot State Abnormal\n");
 else
 printf(" Slot State Normal\n");
 printf(" ASC/ASCQ %02X%02X\n",
 elem->asc, elem->ascq);
 if (elem->full)
 printf(" Media Present Yes\n");
 else
 printf(" Media Present No\n");
 if (elem->access)
 printf(" Robot Access Allowed Yes\n");
 else
 printf(" Robot Access Allowed No\n");
 if (elem->svalid)
 printf(" Source Element Address %d\n", elem->source);
 else
 printf(" Source Element Address Valid ... No\n");
 if (elem->invert)
 printf(" Media Inverted Yes\n");
 else
 printf(" Media Inverted No\n");
 printf(" Volume Tag %0.36s\n", elem->volume);
 printf(" Cartridge Location %0.24s\n", elem->identifier);
}

free(data);
return 0;

```

## Return codes

This chapter describes the return codes that the device driver generates when an error occurs during an operation. The standard *errno* values are in the AIX® `/usr/include/sys/errno.h` header file.

If the return code is input/output error (EIO), the application can issue the **STIOCQRYSENSE** IOCTL command with the LASTERROR option. Or, it can issue the **SIOC\_REQSENSE** IOCTL command to analyze the sense data and determine why the error occurred.

- [Codes for all operations](#)
- [Open error codes](#)
- [Write error codes](#)
- [Read error codes](#)
- [Close error codes](#)
- [IOCTL error codes](#)

---

## Codes for all operations

The following codes and their descriptions apply to all operations.

|                |                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EACCES]       | Data encryption access denied.                                                                                                                                  |
| [EBADF]        | A bad file descriptor was passed to the device.                                                                                                                 |
| [EBUSY]        | An excessive busy state was encountered in the device.                                                                                                          |
| [EFAULT]       | A memory failure occurred due to an invalid pointer or address.                                                                                                 |
| [EMEDIA]       | An unrecoverable media error was detected in the device.                                                                                                        |
| [ENOMEM]       | Insufficient memory was available for an internal memory operation.                                                                                             |
| [ENOTREADY]    | The device was not ready for operation, or a tape was not in the drive.                                                                                         |
| [ENXIO]        | The device was not configured and is not receiving requests.                                                                                                    |
| [EPERM]        | The process does not have permission to complete the wanted function.                                                                                           |
| [ETIMEDOUT]    | A command that is timed out in the device.                                                                                                                      |
| [ENOCCONNECT]  | The device did not respond to selection.                                                                                                                        |
| [ECONNREFUSED] | The device driver detected that the device vital product data (VPD) changed. The device must be unconfigured in AIX® and reconfigured to correct the condition. |

---

## Open error codes

The following codes and their descriptions apply to open operations.

|               |                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EAGAIN]      | The device was opened before the open operation.                                                                                                                                                                        |
| [EBADF]       | A write operation was attempted on a device that was opened with the O_RDONLY flag.                                                                                                                                     |
| [EBUSY]       | The device was reserved by another initiator, or an excessive busy state was encountered.                                                                                                                               |
| [EINVAL]      | The operation that is requested has invalid parameters or an invalid combination of parameters, or the device is rejecting open commands.                                                                               |
| [ENOTREADY]   | If the device was not opened with the O_NONBLOCK or O_NDELAY flag, then the drive is not ready for operation, or a tape is not in the drive. If a nonblocking flag was used, then the drive is not ready for operation. |
| [EWRPROTECT]  | An open operation with the O_RDWR or O_WRONLY flag was attempted on a write-protected tape.                                                                                                                             |
| [EIO]         | An I/O error occurred that indicates a failure to operate the device. Perform the failure analysis.                                                                                                                     |
| [EINPROGRESS] | This <i>errno</i> is returned when the extended open flag SC_KILL_OPEN is used to kill all processes that currently have the device opened.                                                                             |

---

## Write error codes

The following codes and their descriptions apply to write operations.

|          |                                                                                                                                                                                                                                                                                                                                                                                  |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EINVAL] | The operation that is requested has invalid parameters or an invalid combination of parameters.<br>The number of bytes requested in the write operation was not a multiple of the block size for a fixed block transfer.<br><br>The number of bytes requested in the write operation was greater than the maximum block size allowed by the device for variable block transfers. |
| [ENOSPC] | A write operation failed because it reached the early warning mark or the programmable early warning zone (PEWZ) while it was in label-processing mode. This return code is returned only once when the early warning or the programmable early warning zone (PEWZ) is reached.                                                                                                  |
| [ENXIO]  | A write operation was attempted after the device reached the logical end of the medium.                                                                                                                                                                                                                                                                                          |

[EWRPROTECT]

A write operation was attempted on a write-protected tape.

[EIO]

The physical end of the medium was detected, or a general error occurred that indicates a failure to write to the device. Perform the failure analysis.

---

## Read error codes

The following codes and their descriptions apply to read operations.

[EBADF]

A read operation was attempted on a device opened with the O\_WRONLY flag.

[EINVAL]

The operation that is requested has invalid parameters or an invalid combination of parameters.

The number of bytes requested in the read operation was not a multiple of the block size for a fixed block transfer.

The number of bytes requested in the read operation was greater than the maximum size allowed by the device for variable block transfers.

[ENOMEM]

The number of bytes requested in the read operation of a variable block record was less than the size of the block. This error is known as an overlength condition.

---

## Close error codes

The following codes and their descriptions apply to close operations.

[EIO]

An I/O error occurred during the operation. Perform the failure analysis.

[ENOTREADY]

A command that is issued during close, such as a **rewind** command, failed because the device was not ready.

---

## IOCTL error codes

The following codes and their descriptions apply to IOCTL operations.

[EINVAL]

The operation that is requested has invalid parameters or an invalid combination of parameters.

This error code also results if the IOCTL is not supported for the device.

[EWRPROTECT]

An operation that modifies the media was attempted on a write-protected tape or a device opened with the O\_RDONLY flag.

[EIO]

An I/O error occurred during the operation. Perform the failure analysis.

[ECANCELLED]

The **STIOCTOP** IOCTL with the st\_op field that specifies STERASE\_IMM was canceled by another process that issued the **STIOC\_CANCEL\_ERASE** IOCTL.

---

## Linux tape and medium changer device driver

IBM® supplies a tape drive and medium changer device driver for the Linux® platform called IBMTape. IBM also supplies an open source device driver for Linux called lin\_tape. Both IBMTape and lin\_tape have the same programming reference as documented in this manual.

- [Software interface](#)
- [General IOCTL operations](#)
- [Tape drive IOCTL operations](#)
- [Tape drive compatibility IOCTL operations](#)
- [Medium changer IOCTL operations](#)
- [Return codes](#)

---

## Software interface

- [Entry points](#)
- [Medium changer devices](#)

---

## Entry points

IBMTape supports the following Linux-defined entry points.

- [open](#)
- [close](#)
- [read](#)
- [write](#)
- [ioctl](#)

## open

---

This entry point is driven by the open system call.

The programmer can access IBMTape devices with one of 3 access modes: write only, read only, or read and write.

IBMTape also support the append open flag. When the open function is called with the append flag set to TRUE, IBMTape attempts to rewind and seek two consecutive filemarks and place the initial tape position between them. Open append fails [*errno*: EIO] if no tape is loaded or two consecutive filemarks are not on the loaded tape. Open append does not automatically imply write access. Therefore, an access mode must accompany the append flag during the open operation.

The open function issues a SCSI **reserve** command to the target device. If the **reserve** command fails, open fails and *errno* EBUSY is returned.

## close

---

This entry point is driven explicitly by the close system call and implicitly by the operating system at application program termination.

For non-rewinding special files, such as /dev/IBMTape0n, if the last command before the close function was a successful write, IBMTape writes two consecutive filemarks that marks the end of data. It then sets the tape position between the two consecutive filemarks. If the last command before the close function successfully wrote one filemark, then one extra filemark is written that marks the end of data. Then, the tape position is set between the two consecutive filemarks.

For non-rewinding special files, if the last tape command before the close function is write, but the write fails with sense key 6 (Unit Attention) and ASC/ASCQ 29/00 (Power On, Reset, or Bus Device Reset Occurred) or sense key 6 and ASC/ASCQ 28/00 (Not Ready to Ready Transition, Medium May Have Changed), IBMTape does not write two consecutive tape file marks that mark the end of data during close processing. If the last tape command before the close function is write one file mark and that command fails with one of the above two errors, IBMTape does not write one extra file mark that marks the end of data during close processing.

For rewind devices, such as /dev/IBMTape0, if the last command before the close function was a successful write, IBMTape writes two consecutive filemarks that mark the end of data and issues a **rewind** command. If the last command before the close function successfully wrote one filemark, one extra filemark is written marking the end of data, and the **rewind** command is issued. If the **write filemark** command fails, no **rewind** command is issued.

The application writers must be aware that a Unit Attention sense data that is presented means that the tape medium might be in an indeterminate condition, and no assumptions can be made about current tape positioning or whether the medium that was previously in the drive is still in the drive. IBM® suggests that after a Unit Attention is presented, the tape special file be closed and reopened, label processing/verification be run (to determine that the correct medium is mounted), and explicit commands be run to locate to the wanted location. Extra processing might also be needed for particular applications.

If an SIOC\_RESERVE ioctl was issued from an application before close, the close function does not release the device; otherwise, it issues the SCSI release command. In both situations, the close function attempts to deallocate all resources that are allocated for the device. If, for some reason, IBMTape is not able to close, an error code is returned.

Note: The return code for close must always be checked. If close is unsuccessful, retry is recommended.

## read

---

This entry point is driven by the read system call. The read operation can be completed when a tape is loaded in the device.

IBMTape supports two modes of read operation. If the read\_past\_filemark flag is set to TRUE (with the STIOCSETP input/output control [IOCTL]), then when a read operation encounters a filemark, it returns the number of bytes read before it encounters the filemark and sets the tape position after the filemark. If the read\_past\_filemark flag is set to FALSE (by default or with STIOCSETP IOCTL), then when a read operation encounters a filemark, if data was read, the read function returns the number of bytes read, and positions the tape before the filemark. If no data was read, then read returns 0 bytes read and positions the tape after the filemark.

If the read function reaches end of the data on the tape, input/output error (EIO) is returned and ASC, ASCQ keys (obtained by request sense IOCTls) indicate the end of data. IBMTape also conforms to all SCSI standard read operation rules, such as fixed block versus variable block.

## write

---

This entry point is driven by the write system call. The write operation can be completed when a tape is loaded in the device.

IBMTape supports early warning processing. When the trailer\_labels flag is set to TRUE (by default or with STIOCSETP IOCTL call), IBMTape fails with *errno* ENOSPACE only when a write operation first encounters the early warning zone for end of tape. After the ENOSPACE error code is returned, IBMTape suppresses all warning messages from the device that is generated by subsequent write commands, effectively allowing **write** and **write filemark** commands in the early warning zone. When physical end of tape is reached, error code EIO is returned, and the ASC and ASCQ keys (obtained by the request sense IOCTL) confirm the end of physical medium condition. When the trailer\_labels flag is set to FALSE (with STIOCSETP IOCTL call), IBMTape returns the ENOSPACE *errno* when any **write** command is attempted in the early warning zone.

## ioctl

---

This entry point provides a set of drive SCSI-specific functions. It allows Linux applications to access and control the features and attributes of the drive device programmatically.

## Medium changer devices

---

IBMTape supports the following Linux® entry points for the medium changer devices.



- [open](#)
- [close](#)
- [ioctl](#)

## open

---

This entry point is driven by the open system call. The open function attempts a SCSI **reserve** command to the target device. If the **reserve** command fails, open fails with *errno* EBUSY.

## close

---

This entry point is driven explicitly by the close system call and implicitly by the operating system at program termination. If an SIOC\_RESERVE IOCTL was issued from an application before close, the close function does not release the device. Otherwise, it issues the SCSI **release** command. In both situations, the close function attempts to deallocate all resources that are allocated for the device. If, for some reason, IBMtape is not able to close, an error code is returned.

## ioctl

---

This entry point provides a set of medium changer and SCSI-specific functions. It allows Linux applications to access and control the features and attributes of the robotic device programmatically.

## General IOCTL operations

---

This chapter describes the IOCTL commands that provide access and control to the tape and medium changer devices.

These commands are available for all tape and medium changer devices. They can be issued to any one of the IBMtape special files.

- [Overview](#)

## Overview

---

The following IOCTL commands are supported.

SIOC\_INQUIRY  
Return the inquiry data.

SIOC\_REQSENSE  
Return the sense data.

SIOC\_RESERVE  
Reserve the device.

SIOC\_RELEASE  
Release the device.

SIOC\_TEST\_UNIT\_READY  
Issue the **SCSI Test Unit Ready** command.

SIOC\_LOG\_SENSE\_PAGE  
Return the log sense data.

SIOC\_LOG\_SENSE10\_PAGE  
Return the log sense data by using a 10-byte CDB with optional subpage.

SIOC\_ENH\_LOG\_SENSE  
Return the page data with a requested length from the application if no kernel memory restriction exists.

SIOC\_MODE\_SENSE\_PAGE  
Return the mode sense data.

SIOC\_MODE\_SENSE  
Return the mode sense data with optional subpage.

SIOC\_INQUIRY\_PAGE  
Return the inquiry data for a specific page.

SIOC\_PASS\_THROUGH  
Pass through custom built SCSI commands.

SIOC\_QUERY\_PATH  
Return the primary path and information for the first alternate path.

SIOC\_DEVICE\_PATHS  
Return the primary path and information for all the alternate paths.

SIOC\_ENABLE\_PATH  
Enable a path from the disabled state.

SIOC\_DISABLE\_PATH  
Disable a path.

These IOCTL commands and their associated structures are defined in the IBM\_tape.h header file, which can be found in /usr/include/sys after IBMtape is installed. The IBM\_tape.h header file must be included in the corresponding C programs that call functions that are provided by IBMtape.

All IOCTL commands require a file descriptor of an open file. Use the **open** command to open a device and obtain a valid file descriptor.

- [SIOC\\_INQUIRY](#)
- [SIOC\\_REQSENSE](#)
- [SIOC\\_RESERVE](#)
- [SIOC\\_RELEASE](#)

- [SIOC\\_TEST\\_UNIT\\_READY](#)
- [SIOC\\_LOG\\_SENSE\\_PAGE, SIOC\\_LOG\\_SENSE10\\_PAGE, and SIOC\\_ENH\\_LOG\\_SENSE](#)
- [SIOC\\_MODE\\_SENSE\\_PAGE and SIOC\\_MODE\\_SENSE](#)
- [SIOC\\_INQUIRY\\_PAGE](#)
- [SCSI\\_PASS\\_THROUGH](#)
- [SIOC\\_QUERY\\_PATH](#)
- [SIOC\\_DEVICE\\_PATHS](#)
- [SIOC\\_ENABLE\\_PATH](#)
- [SIOC\\_DISABLE\\_PATH](#)

## SIOC\_INQUIRY

This IOCTL command collects the inquiry data from the device.

The data structure is

```
struct inquiry_data {
 uint qual :3, /* peripheral qualifier */
 type :5, /* device type */
 uint rm :1, /* removable medium */
 mod :7, /* device type modifier */
 uint iso :2, /* ISO version */
 ecma :3, /* EMCA version */
 ansi :3, /* ANSI version */
 uint aenc :1, /* asynchronous event notification */
 trmiop :1, /* terminate I/O process message */
 reserved :2, /* reserved */
 rdf :4, /* response data format */
 uchar len; /* additional length */
 uchar resvd1; /* reserved */
 uint reserved :4, /* reserved */
 mchngr :1, /* medium changer mode (SCSI-3 only) */
 reserved :3, /* reserved */
 uint reladr :1, /* relative addressing */
 wbus32 :1, /* 32-bit wide data transfers */
 wbus16 :1, /* 16-bit wide data transfers */
 sync :1, /* synchronous data transfers */
 linked :1, /* linked commands */
 reserved :1, /* reserved */
 cmdque :1, /* command queueing */
 sftre :1, /* soft reset */
 uchar vid[8]; /* vendor ID */
 uchar pid[16]; /* product ID */
 uchar revision[4]; /* product revision level */
 uchar vendor1[20]; /* vendor specific */
 uchar resvd2[40]; /* reserve */
 uchar vendor2[31]; /* vendor specific (padded to 127) */
};
```

An example of the **SIOC\_INQUIRY** command is

```
#include <sys/IBM_tape.h>
char vid[9];
char pid[17];
char revision[5];
struct inquiry_data inqdata;
printf("Issuing inquiry...\n");
memset(&inqdata, 0, sizeof(struct inquiry_data));
if (!ioctl (fd, SIOC_INQUIRY, &inqdata)) {
 printf ("The SIOC_INQUIRY ioctl succeeded\n");
 printf ("\nThe inquiry data is:\n");
 /*-
 * Just a dump byte won't work because of the compiler
 * bit field mapping
 */
 /* print out structure data field */
 printf("\nInquiry Data:\n");
 printf("Peripheral Qualifier-----0x%02x\n", inqdata.qual);
 printf("Peripheral Device Type-----0x%02x\n", inqdata.type);
 printf("Removal Medium Bit-----%d\n", inqdata.rm);
 printf("Device Type Modifier-----0x%02x\n", inqdata.mod);
 printf("ISO version-----0x%02x\n", inqdata.iso);
 printf("ECMA version-----0x%02x\n", inqdata.ecma);
 printf("ANSI version-----0x%02x\n", inqdata.ansi);
 printf("Asynchronous Event Notification Bit-%d\n", inqdata.aenc);
 printf("Terminate I/O Process Message Bit---%d\n", inqdata.trmiop);
 printf("Response Data Format-----0x%02x\n", inqdata.rdf);
 printf("Additional Length-----0x%02x\n", inqdata.len);
 printf("Medium Changer Mode-----0x%02x\n", inqdata.mchngr);
 printf("Relative Addressing Bit-----%d\n", inqdata.reladr);
 printf("32 Bit Wide Data Transfers Bit-----%d\n", inqdata.wbus32);
 printf("16 Bit Wide Data Transfers Bit-----%d\n", inqdata.wbus16);
 printf("Synchronous Data Transfers Bit-----%d\n", inqdata.sync);
 printf("Linked Commands Bit-----%d\n", inqdata.linked);
 printf("Command Queueing Bit-----%d\n", inqdata.cmdque);
 printf("Soft Reset Bit-----%d\n", inqdata.sftre);

 strncpy(vid, inqdata.vid, 8);
 vid[8] = '\0';
 strncpy(pid, inqdata.pid, 16);
```

```

 pid[16] = '\0';
 strncpy(revision, inqdata.revision, 4);
 revision[4] = '\0';

 printf("Vendor ID-----%s\n", vid);
 printf("Product ID-----%s\n", pid);
 printf("Product Revision Level-----%s\n", revision);

 dump_bytes(inqdata.vendor1, 20, "vendor1");
 dump_bytes(inqdata.vendor2, 31, "vendor2");
}
else {
 perror ("The SIOC_INQUIRY ioctl failed");
 sioc_request_sense();
}

```

## SIOC\_REQSENSE

This IOCTL command returns the device sense data. If the last command resulted in an error, then the sense data is returned for the error. Otherwise, a new sense command is issued to the device.

The data structure is

```

struct request_sense {
 uint valid :1, /* sense data is valid */
 uint err_code :7, /* error code */
 uchar segnum; /* segment number */
 uint fm :1, /* filemark detected */
 uint eom :1, /* end of medium */
 uint ili :1, /* incorrect length indicator */
 uint resvdl :1, /* reserved */
 uint key :4, /* sense key */
 int info; /* information bytes */
 uchar addlen; /* additional sense length */
 uint cmdinfo; /* command specific information */
 uchar asc; /* additional sense code */
 uchar ascq; /* additional sense code qualifier */
 uchar fru; /* field replaceable unit code */
 uint sksv :1, /* sense key specific valid */
 uint cd :1, /* control/data */
 uint resvd2 :2, /* reserved */
 uint bpv :1, /* bit pointer valid */
 uint sim :3, /* system information message */
 uchar field[2]; /* field pointer */
 uchar vendor[109]; /* vendor specific (padded to 127) */
};

```

An example of the **SIOC\_REQSENSE** command is

```

#include <sys/IBM_tape.h>

struct request_sense sense_data;
int rc;
printf("Issuing request sense...\n");
memset(&sense_data, 0, sizeof(struct request_sense));
rc = ioctl(fd, SIOC_REQSENSE, &sense_data);
if (rc == 0)
{
 if(!sense_data.err_code)
 printf("No valid sense data returned.\n");
 else
 {
 /* print out data fields */
 printf("Information Field Valid Bit-----%d\n", sense_data.valid);
 printf("Error Code-----0x%02x\n", sense_data.err_code);
 printf("Segment Number-----0x%02x\n", sense_data.segnum);
 printf("filemark Detected Bit-----%d\n", sense_data.fm);
 printf("End Of Medium Bit-----%d\n", sense_data.eom);
 printf("Illegal Length Indicator Bit----%d\n", sense_data.ili);
 printf("Sense Key-----0x%02x\n", sense_data.key);
 if(sense_data.valid)
 printf("Information Bytes-----0x%02x 0x%02x 0x%02x 0x%02x\n",
 sense_data.info >> 24, sense_data.info >> 16,
 sense_data.info >> 8, sense_data.info & 0xFF);
 printf("Additional Sense Length-----0x%02x\n", sense_data.addlen);
 printf("Command Specific Information---0x%02x 0x%02x 0x%02x 0x%02x\n",
 sense_data.cmdinfo >> 24, sense_data.cmdinfo >> 16,
 sense_data.cmdinfo >> 8, sense_data.cmdinfo & 0xFF);
 printf("Additional Sense Code-----0x%02x\n", sense_data.asc);
 printf("Additional Sense Code Qualifier-0x%02x\n", sense_data.ascq);
 printf("Field Replaceable Unit Code-----0x%02x\n", sense_data.fru);
 printf("Sense Key Specific Valid Bit----%d\n", sense_data.sksv);
 if(sense_data.sksv)
 {
 printf("Command Data Block Bit--%d\n", sense_data.cd);
 printf("Bit Pointer Valid Bit---%d\n", sense_data.bpv);
 if(sense_data.bpv)
 printf("System Information Message-0x%02x\n", sense_data.sim);
 printf("Field Pointer-----0x%02x%02x\n",
 sense_data.field[0], sense_data.field[1]);
 }
 }
}

```

```

 dump_bytes(sense_data.vendor, 109, "Vendor");
}
}
return rc;

```

---

## SIOC\_RESERVE

This IOCTL command explicitly reserves the device and prevents it from being released after a close operation.

The device is not released until an **SIOC\_RELEASE** IOCTL command is issued.

The IOCTL command can be used for applications that require multiple open and close processing in a host-sharing environment.

There are no arguments for this IOCTL command.

An example of the **SIOC\_RESERVE** command is

```

#include <sys/IBM_tape.h>
if (!ioctl (fd, SIOC_RESERVE, NULL)) {
 printf ("The SIOC_RESERVE ioctl succeeded\n");
}
else {
 perror ("The SIOC_RESERVE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_RELEASE

This IOCTL command explicitly releases the device and allows other hosts to access it. The IOCTL command is used with the **SIOC\_RESERVE** IOCTL command for applications that require multiple open and close processing in a host-sharing environment.

There are no arguments for this IOCTL command.

An example of the **SIOC\_RELEASE** command is

```

#include <sys/IBM_tape.h>
if (!ioctl (fd, SIOC_RELEASE, NULL)) {
 printf ("The SIOC_RELEASE ioctl succeeded\n");
}
else {
 perror ("The SIOC_RELEASE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_TEST\_UNIT\_READY

This IOCTL command issues the **SCSI Test Unit Ready** command to the device.

There are no arguments for this IOCTL command.

An example of the **SIOC\_TEST\_UNIT\_READY** command is

```

#include <sys/IBM_tape.h>
if (!ioctl (fd, SIOC_TEST_UNIT_READY, NULL)) {
 printf ("The SIOC_TEST_UNIT_READY ioctl succeeded\n");
}
else {
 perror ("The SIOC_TEST_UNIT_READY ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_LOG\_SENSE\_PAGE, SIOC\_LOG\_SENSE10\_PAGE, and SIOC\_ENH\_LOG\_SENSE

These IOCTL commands return log sense data from the device. The differences between the three is

- **SIOC\_LOG\_SENSE\_PAGE** allows the user to retrieve a particular log page up to length LOGSENSEPAGE.
- **SIOC\_LOG\_SENSE10\_PAGE** allows for a subpage to be returned up to length LOGSENSEPAGE.
- **SIOC\_ENH\_LOG\_SENSE** returns the page data with a requested length from application if no kernel memory restriction exists.

For both **SIOC\_LOG\_SENSE\_PAGE** and **SIOC\_LOG\_SENSE10\_PAGE**, to obtain the entire log page, the len and parm\_pointer fields must be set to zero. To obtain the entire log page that starts at a specific parameter code, set the parm\_pointer field to the wanted code and the len field to zero. To obtain a specific number of parameter bytes, set the parm\_pointer field to the wanted code. Then, set the len field to the number of parameter bytes plus the size of the log page header (4 bytes). The first 4 bytes of returned data are always the log page header. In the Enhanced log sense page (**SIOC\_ENH\_LOG\_SENSE**), the length cannot be set to zero as it indicates the allocated memory size that char \*logdatap is pointing to. The minimum number for this value is 4, as it returns the first 4 bytes that is the log page header. See the appropriate device manual to determine the supported log pages and content.

The data structures are

```

struct log_sense_page {
 uchar page_code;
 unsigned short len;
 unsigned short parm_pointer;
 char data[LOGSENSEPAGE];
};

struct log_sense10_page {
 uchar page_code;
 uchar subpage_code;
 uchar reserved[2];
 unsigned short len;
 unsigned short parm_pointer;
 char data[LOGSENSEPAGE];
};

struct enh_log_sense {
 uchar page_code; /* [IN] Log sense page */
 uchar subpage_code; /* [IN] Log sense sub-page */
 uchar page_control; /* [IN] Page control */
 uchar reserved[5];
 unsigned short len; /* [IN] specific allocation length for logdatap */
 /* by application */
 /* [OUT] the length of return data at */
 /* logdatap from driver */
 unsigned short parm_pointer; /* [IN] specific parameter number at */
 /* which the data begins */
 char *logdatap; /* [IN] the pointer for log sense data allocated */
 /* by application */
 /* [OUT] log sense data returned from driver */
};

```

The first two IOCTLs are identical, except if a specific subpage is wanted, `log_sense10_page` must be used and `subpage_code` must be assigned by the user application.

An example of the **SIOC\_LOG\_SENSE\_PAGE** command is

```

#include <sys/IBM_tape.h>
struct log_sense_page log_page;
int temp;
/* get log page 0, list of log pages */
log_page.page_code = 0x00;
log_page.len = 0;
log_page.parm_pointer = 0;
if (!ioctl (fd, SIOC_LOG_SENSE_PAGE, &log_page)) {
 printf ("The SIOC_LOG_SENSE_PAGE ioctl succeeded\n");
 dump_bytes(log_page.data, LOGSENSEPAGE);
}
else {
 perror ("The SIOC_LOG_SENSE_PAGE ioctl failed");
 sioc_request_sense();
}
/* get fraction of volume traversed */
log_page.page_code = 0x38;
log_page.len = 0;
log_page.parm_pointer = 0x000F;
if (!ioctl (fd, SIOC_LOG_SENSE_PAGE, &log_page)) {
 temp = log_page.data[sizeof(log_page_header) + 4];
 printf ("The SIOC_LOG_SENSE_PAGE ioctl succeeded\n");
 printf ("Fractional Part of Volume Traversed %x\n",temp);
}
else {
 perror ("The SIOC_LOG_SENSE_PAGE ioctl failed");
 sioc_request_sense();
}

```

An example of the **SIOC\_ENH\_LOG\_SENSE** command is

```

include <sys/IBM_tape.h>

#define LOG_PAGE_HEADER 4

struct enh_log_sense enh_log_page;
unsigned short length;

memset((char*)&enh_log_page, 0, sizeof(struct enh_log_sense));
enh_log_page.page_code = 0x17;
enh_log_page.subpage_code = 0x02;
enh_log_page.len = LOG_PAGE_HEADER;
enh_log_page.logdatap = malloc(LOG_PAGE_HEADER);

if(enh_log_page.logdatap == NULL){
 printf ("Unable to malloc LOG_PAGE_HEADER. Closing\n");
 exit(-1);
}

if (!ioctl (fd, SIOC_ENH_LOG_SENSE, &enh_log_page)) {
 printf ("The SIOC_ENH_LOG_SENSE ioctl succeeded\n");
 sprintf(text,"Log enhanced page header 0x%02X subpage 0x%02X",
 enh_log_page.page_code, enh_log_page.subpage_code);
 dump_bytes(enh_log_page.logdatap, enh_log_page.len, text);
}
else {
 perror ("The SIOC_ENH_LOG_SENSE ioctl failed");
}

```

```

}

length = (enh_log_page.logdatap[2] << 8) + enh_log_page.logdatap[3];
free(enh_log_page.logdatap);
enh_log_page.logdatap = NULL;

enh_log_page.len = length;
enh_log_page.logdatap = malloc(length);

if(enh_log_page.logdatap==NULL) {
 printf("Unable to malloc enh_log_page big size %d\n", length);
 if(length > 1024) {
 enh_log_page.logdatap = malloc(1024);
 enh_log_page.len = 1024;
 if(enh_log_page.logdatap == NULL) {
 printf("Unable to malloc enh_log_page 1024 size\n");
 exit(-1);
 }
 }
}

if (!ioctl (fd, SIOC_ENH_LOG_SENSE, &enh_log_page)) {
 printf ("The SIOC_ENH_LOG_SENSE ioctl succeeded\n");
 sprintf(text, "Enhanced Log Sense: page 0x%02X subpage 0x%02X length %d",
 enh_log_page.page_code, enh_log_page.subpage_code);

 dump_bytes(enh_log_page.logdatap, enh_log_page.len, text);
}
else {
 perror ("The SIOC_ENH_LOG_SENSE ioctl failed");
}
free(enh_log_page.logdatap);

```

---

## SIOC\_MODE\_SENSE\_PAGE and SIOC\_MODE\_SENSE

This IOCTL command returns a mode sense page from the device. The desired page is selected by specifying the `page_code` in the `mode_sense_page` structure. See the appropriate device manual to determine the supported mode pages and content.

The data structures are

```

struct mode_sense_page {
 unchar page_code;
 char data[MAX_MDSNS_LEN];
};

struct mode_sense {
 unchar page_code;
 unchar subpage_code;
 unchar reserved[6];
 unchar cmd_code;
 char data[MAX_MDSNS_LEN];
};

```

The IOCTLs are identical, except that if a specific subpage is desired, `mode_sense` must be used and `subpage_code` must be assigned by the user application. Under the current implementation, `cmd_code` is not assigned by the user and must be left with a value 0.

An example of the **SIOC\_MODE\_SENSE\_PAGE** command is

```

#include <sys/IBM_tape.h>
struct mode_sense_page mode_page;
/* get medium changer mode */
mode_page.page_code = 0x20;
if (!ioctl (fd, SIOC_MODE_SENSE_PAGE, &mode_page)) {
 printf ("The SIOC_MODE_SENSE_PAGE ioctl succeeded\n");
 if (mode_page.data[2] == 0x02)
 printf ("The library is in Random mode.\n");
 else if (mode_page.data[2] == 0x05)
 printf ("The library is in Automatic (Sequential) mode.\n");
}
else {
 perror ("The SIOC_MODE_SENSE_PAGE ioctl failed");
 sioc_request_sense();
}

```

---

## SIOC\_INQUIRY\_PAGE

This IOCTL command returns an inquiry page from the device. The desired page is selected by specifying the `page_code` in the `inquiry_page` structure. See the appropriate device manual to determine the supported inquiry pages and content.

The data structure is

```

struct inquiry_page {
 char page_code;
 char data[INQUIRYPAGE];
};

```

An example of the **SIOC\_INQUIRY\_PAGE** command is

```
#include <sys/IBM_tape.h>
struct inquiry_page inq_page;
/* get inquiry page x83 */
inq_page.page_code = 0x83;
if (!ioctl (fd, SIOC_INQUIRY_PAGE, &inq_page)) {
 printf ("The SIOC_INQUIRY_PAGE ioctl succeeded\n");
 dump_bytes(inq_page.data, INQUIRYPAGE);
}
else {
 perror ("The SIOC_INQUIRY_PAGE ioctl failed");
 sioc_request_sense();
}
```

## SCSI\_PASS\_THROUGH

This IOCTL command passes the built command data block structure with I/O buffer pointers to the lower SCSI layer. Status is returned from the lower SCSI layer to the caller with the ASC and ASCQ values and SenseKey fields. The ASC and ASCQ and sense key fields are valid only when the SenseDataValid field is true.

The data structure is

```
#define SCSI_PASS_THROUGH_IOWR('P',0x01,SCSIPassThrough) /* Pass Through */
typedef struct _SCSIPassThrough
{
 unchar CDB[12]; /* Command Data Block */
 unchar CommandLength; /* Command Length */
 unchar * Buffer; /* Command Buffer */
 ulong BufferLength; /* Buffer Length */
 unchar DataDirection; /* Data Transfer Direction */
 ushort Timeout; /* Time Out Value */
 unchar TargetStatus; /* Target Status */
 unchar MessageStatus; /* Message from host adapter */
 unchar HostStatus; /* Host status */
 unchar DriverStatus; /* Driver status */
 unchar SenseDataValid; /* Sense Data Valid */
 unchar ASC; /* ASC key if the SenseDataValid is True */
 unchar ASCQ; /* ASCQ key if the SenseDataValid is True */
 unchar SenseKey; /* Sense key if the SenseDataValid is True */
} SCSIPassThrough, *PSCSIPassThrough;
#define SCSI_DATA_OUT 1
#define SCSI_DATA_IN 2
#define SCSI_DATA_NONE 3
```

SCSI\_DATA\_OUT indicates sending data out of the initiator (host bus adapter), also known as write mode. SCSI\_DATA\_IN indicates receiving data into the initiator (host bus adapter), also known as read mode. SCSI\_DATA\_NONE indicates that no data is transferred.

An example of the SCSI\_PASS\_THROUGH command is

```
#include <sys/IBM_tape.h>
SCSIPassThrough PassThrough;
memset(&PassThrough, 0, sizeof(SCSIPassThrough));
/* Issue test unit ready command */
PassThrough.CDB[0] = 0x00;
PassThrough.CommandLength = 6;
PassThrough.DataDirection = SCSI_DATA_NONE;
if (!ioctl (fd, SCSI_PASS_THROUGH, &PassThrough)) {
 printf ("The SCSI_PASS_THROUGH ioctl succeeded\n");
 if ((PassThrough.TargetStatus == STATUS_SUCCESS) &&
 (PassThrough.MessageStatus == STATUS_SUCCESS) &&
 (PassThrough.HostStatus == STATUS_SUCCESS) &&
 (PassThrough.DriverStatus == STATUS_SUCCESS))
 printf(" Test Unit Ready returns success\n");
}
else {
 printf(" Test Unit Ready failed\n");
 if(PassThrough.SenseDataValid)
 printf("Sense Key %02x, ASC %02x, ASCQ %02x\n",
 PassThrough.SenseKey, PassThrough.ASC,
 PassThrough.ASCQ);
}
}
else {
 perror ("The SIOC SCSI_PASS_THROUGH ioctl failed");
 sioc_request_sense();
}
```

## SIOC\_QUERY\_PATH

This IOCTL command returns the primary path and the first alternate path information for a physical device.

The data structure is

```
struct scsi_path
{
 char primary_name[30]; /* primary logical device name */
 char primary_parent[30]; /* primary SCSI parent name, "Host" name */
 unchar primary_id; /* primary target address of device, "Id" value*/
 unchar primary_lun; /* primary logical unit of device, "lun" value */
}
```

```

 unchar primary_bus; /* primary SCSI bus for device, "Channel" value*/
 unsigned long long primary_fcp_scsi_id; /* not supported */
 unsigned long long primary_fcp_lun_id; /* not supported */
 unsigned long long primary_fcp_ww_name; /* not supported */
 unchar primary_enabled; /* primary path enabled */
 unchar primary_id_valid; /* primary id/lun/bus fields valid */
 unchar primary_fcp_id_valid; /* not supported */
 unchar alternate_configured; /* alternate path configured */
 char alternate_name[30]; /* alternate logical device name */
 char alternate_parent[30]; /* alternate SCSI parent name */
 unchar alternate_id; /* alternate target address of device */
 unchar alternate_lun; /* alternate logical unit of device */
 unchar alternate_bus; /* alternate SCSI bus for device */
 unsigned long long alternate_fcp_scsi_id; /* not supported */
 unsigned long long alternate_fcp_lun_id; /* not supported */
 unsigned long long alternate_fcp_ww_name; /* not supported */
 unchar alternate_enabled; /* alternate path enabled */
 unchar alternate_id_valid; /* alternate id/lun/bus fields valid */
 unchar alternate_fcp_id_valid; /* not supported */
 unchar primary_drive_port_valid; /* not supported */
 unchar primary_drive_port; /* not supported */
 unchar alternate_drive_port_valid; /* not supported */
 unchar alternate_drive_port; /* not supported */
 unchar primary_fenced; /* primary fenced by disable path ioctl */
 unchar alternate_fenced; /* alternate fenced by disable path ioctl */
 unchar primary_host; /* primary host bus adapter id */
 unchar alternate_host; /* alternate host bus adapter id */
 char reserved[56];
};

```

An example of the **SIOC\_QUERY\_PATH** command is

```

#include <sys/IBM_tape.h>
struct scsi_path path;
memset(&path, 0, sizeof(struct scsi_path));
printf("Querying SCSI paths...\n");
rc = ioctl(fd, SIOC_QUERY_PATH, &path);
if(rc == 0)
 show_path(&path);

```

## SIOC\_DEVICE\_PATHS

This IOCTL command returns the primary path and all of the alternate paths information for a physical device. This IOCTL supports only the 3592 tape drives. The data structure for this IOCTL command is

```

struct device_path_t
{
 char name[30]; /* logical device name */
 char parent[30]; /* logical parent name */
 unchar id_valid; /* SCSI id/lun/bus fields valid */
 unchar id; /* SCSI target address of device */
 unchar lun; /* SCSI logical unit of device */
 unchar bus; /* SCSI bus for device */
 unchar fcp_id_valid; /* not supported */
 unsigned long long fcp_scsi_id; /* not supported */
 unsigned long long fcp_lun_id; /* not supported */
 unsigned long long fcp_ww_name; /* not supported */
 unchar enabled; /* path enabled */
 unchar drive_port_valid; /* not supported */
 unchar drive_port; /* not supported */
 unchar fenced; /* path fenced by diable path ioctl */
 unchar host; /* host bus adapter id */
 char reserved[62];
};

struct device_paths
{
 int number_paths; /* number of paths configured */
 struct device_path_t path[MAX SCSI_PATH];
};

```

An example of this IOCTL command is

```

#include <sys/IBM_tape.h>
struct device_paths device_path;
memset(&device_path, 0, sizeof(struct device_paths));
printf("Querying device paths...\n");
rc = ioctl(fd, SIOC_DEVICE_PATHS, &device_path);
if(rc == 0)
{
 printf("\n");
 for (i=0; i < device_path.number_paths; i++)
 {
 if (i == 0)
 printf("Primary Path Number 1\n");
 else
 printf("Alternate Path Number %d\n", i+1);
 printf(" Logical Device..... %s\n", device_path.path[i].name);
 printf(" Host Bus Adapter..... %s\n", device_path.path[i].parent);

 if (device_path.path[i].id_valid)

```



```

 {
 printf(" SCSI Host ID..... %d\n",device_path.path[i].host);
 printf(" SCSI Channel..... %d\n",device_path.path[i].bus);
 printf(" Target ID..... %d\n",device_path.path[i].id);
 printf(" Logical Unit..... %d\n",device_path.path[i].lun);
 }

 if (device_path.path[i].enabled)
 printf(" Path Enabled..... Yes\n");
 else
 printf(" Path Enabled..... No \n");
 if (device_path.path[i].fenced)
 printf(" Path Manually Disabled..... Yes\n");
 else
 printf(" Path Manually Disabled..... No \n");

 printf("\n");
 }

 printf("Total paths configured..... %d\n",device_path.number_paths);
}

```

---

## SIOC\_ENABLE\_PATH

This IOCTL enables the path that is specified by the path number. This command supports only the 3592 tape drives.

An example of this IOCTL command is

```

#include <sys/IBM_tape.h>
if (path == PRIMARY_SCSI_PATH)
 printf("Enabling primary SCSI path 1...\n");
else
 printf("Enabling alternate SCSI path %d...\n",path);

rc = ioctl(fd, SIOC_ENABLE_PATH, path);

```

---

## SIOC\_DISABLE\_PATH

This IOCTL disables the path that is specified by the path number. This command supports only the 3592 tape drives.

An example of this IOCTL command is

```

#include <sys/IBM_tape.h>
if (path == PRIMARY_SCSI_PATH)
 printf("Disabling primary SCSI path 1...\n");
else
 printf("Disabling alternate SCSI path %d...\n",path);

rc = ioctl(fd, SIOC_DISABLE_PATH, path);

```

---

## Tape drive IOCTL operations

The device driver supports the set of tape IOCTL commands that is available with the base Linux® operating system. In addition, a set of expanded tape IOCTL commands gives applications access to extra features and functions of the tape drives.

- [Overview](#)

---

## Overview

The following IOCTL commands are supported.

```

STIOCTOP
 Run the basic tape operations.
STIOCQRYP
 Query the tape device, device driver, and media parameters.
STIOCSETP
 Change the tape device, device driver, and media parameters.
STIOCSYNC
 Synchronize the tape buffers with the tape.
STIOCDM
 Displays and manipulates one or two messages.
STIOCQRYPOS
 Query the tape position and the buffered data.
STIOCSETPOS
 Set the tape position.
STIOCQRYSENSE
 Query the sense data from the tape device.

```

**STIOCQRYINQUIRY**  
Return the inquiry data.

**STIOC\_LOCATE**  
Locate to a certain tape position.

**STIOC\_READ\_POSITION**  
Read the current tape position.

**STIOC\_RESET\_DRIVE**  
Issue a **SCSI Send Diagnostic** command to reset the tape drive.

**STIOC\_PREVENT\_MEDIUM\_REMOVAL**  
Prevent medium removal by an operator.

**STIOC\_ALLOW\_MEDIUM\_REMOVAL**  
Allow medium removal by an operator.

**STIOC\_REPORT\_DENSITY\_SUPPORT**  
Return supported densities from the tape device.

**MTDEVICE**  
Returns the device number that is used for communicating with an Enterprise Tape Library 3494.

**STIOC\_GET\_DENSITY**  
Query the current write density format settings on the tape drive. The current density code from the drive Mode Sense header, the Read/Write Control Mode page default density, and the pending density are returned.

**STIOC\_SET\_DENSITY**  
Set a new write density format on the tape drive by using the default and pending density fields. The application can specify a new write density for the currently loaded tape only. Or, it can specify a new write density as a default for all tapes.

**GET\_ENCRYPTION\_STATE**  
This IOCTL can be used for application, system, and library-managed encryption. It allows only a query of the encryption status.

**SET\_ENCRYPTION\_STATE**  
This IOCTL can be used only for application-managed encryption. It sets the encryption state for application-managed encryption.

**SET\_DATA\_KEY**  
This IOCTL can be used only for application-managed encryption. It sets the data key for application-managed encryption.

**STIOC\_QUERY\_PARTITION**  
This IOCTL queries for partition information on applicable tapes. It displays maximum number of possible partitions, number of partitions currently on tape, the active partition, the size unit (bytes, kilobytes, and so on), and the sizes of each partition.

**STIOC\_CREATE\_PARTITION**  
This IOCTL creates partitions on applicable tapes. The user is allowed to specify the number and type of partitions and the size of each partition.

**STIOC\_SET\_ACTIVE\_PARTITION**  
This IOCTL allows the user to set the partition on which to complete tape operations.

**STIOC\_ALLOW\_DATA\_OVERWRITE**  
This IOCTL allows tape data to be overwritten when in data safe mode.

**STIOC\_READ\_POSITION\_EX**  
This IOCTL reads the tape position and includes support for the long and extended formats.

**STIOC\_LOCATE\_16**  
This IOCTL sets the tape position by using a long tape format.

**STIOC\_QUERY\_BLK\_PROTECTION**  
This IOCTL queries the current capability and status of Logical Block Protection in the drive.

**STIOC\_SET\_BLK\_PROTECTION**  
This IOCTL sets the status of Logical Block Protection in the drive.

**STIOC\_VERIFY\_TAPE\_DATA**  
This IOCTL instructs the tape drive to scan the data on its current tape to check for errors.

**STIOC\_QUERY\_RAO**  
The IOCTL is used to query the maximum number and size of User Data Segments (UDS) that are supported from tape drive and driver for the wanted uds\_type.

**STIOC\_GENERATE\_RAO**  
The IOCTL is called to send a GRAO list to request that the drive generate a Recommended Access Order list.

**STIOC\_RECEIVE\_RAO**  
After a **STIOC\_GENERATE\_RAO** IOCTL is completed, the application calls the **STIOC\_RECEIVE\_RAO** IOCTL to receive a recommended access order of UDS from the drive.

**STIOC\_SET\_SPDEV**  
This IOCTL is for usage through IBMSpecial open handle only. It sets the drive that processes the command requests, and to do so it needs the serial number of the drive as input.

These IOCTL commands and their associated structures are defined in the `IBM_tape.h` header file that can be found in the `lin_tape` source rpm package. This header must be included in the corresponding C program by using the IOCTL commands.

- [\*\*STIOCTOP\*\*](#)
- [\*\*STIOCQRYP or STIOCSETP\*\*](#)
- [\*\*STIOCSYNC\*\*](#)
- [\*\*STIOCDM\*\*](#)
- [\*\*STIOCQRYPOS\*\*](#)
- [\*\*STIOCSETPOS\*\*](#)
- [\*\*STIOCQRYSENSE\*\*](#)
- [\*\*STIOCQRYINQUIRY\*\*](#)
- [\*\*STIOC\\_LOCATE\*\*](#)
- [\*\*STIOC\\_READ\\_POSITION\*\*](#)
- [\*\*STIOC\\_RESET\\_DRIVE\*\*](#)
- [\*\*STIOC\\_PREVENT\\_MEDIUM\\_REMOVAL\*\*](#)
- [\*\*STIOC\\_ALLOW\\_MEDIUM\\_REMOVAL\*\*](#)
- [\*\*STIOC\\_REPORT\\_DENSITY\\_SUPPORT\*\*](#)
- [\*\*MTDEVICE \(Obtain Device Number\)\*\*](#)
- [\*\*STIOC\\_GET\\_DENSITY and STIOC\\_SET\\_DENSITY\*\*](#)
- [\*\*GET\\_ENCRYPTION\\_STATE\*\*](#)
- [\*\*SET\\_ENCRYPTION\\_STATE\*\*](#)
- [\*\*SET\\_DATA\\_KEY\*\*](#)

- [STIOC\\_QUERY\\_PARTITION](#)
- [STIOC\\_CREATE\\_PARTITION](#)
- [STIOC\\_SET\\_ACTIVE\\_PARTITION](#)
- [STIOC\\_ALLOW\\_DATA\\_OVERWRITE](#)
- [STIOC\\_READ\\_POSITION\\_EX](#)
- [STIOC\\_LOCATE\\_16](#)
- [STIOC\\_QUERY\\_BLK\\_PROTECTION](#)
- [STIOC\\_SET\\_BLK\\_PROTECTION](#)
- [STIOC\\_VERIFY\\_TAPE\\_DATA](#)
- [STIOC\\_QUERY\\_RAO](#)
- [STIOC\\_GENERATE\\_RAO](#)
- [STIOC\\_RECEIVE\\_RAO](#)
- [STIOC\\_SET\\_SPDEV](#)

---

## STIOCTOP

This IOCTL command runs basic tape operations. The `st_count` variable is used for many of its operations. Normal error recovery applies to these operations. The device driver can issue several tries to complete them. For all forward movement space operations, the tape position finishes on the end-of-tape side of the record or filemark, and on the beginning-of-tape side of the record or filemark for backward movement.

The input data structure is

```
struct stop {
 short st_op; /* operations defined below */
 daddr_t st_count; /* how many of them to do (if applicable) */
};
```

The `st_op` variable is set to one of the following operations.

### STOFFL

Unload the tape. The `st_count` parameter does not apply.

### STREW

Rewind the tape. The `st_count` parameter does not apply.

### STERASE

Erase the entire tape. The `st_count` parameter does not apply.

### STRETEN

Run the rewind operation. The tape devices run the retension operation automatically when needed.

### STWEOF

Write `st_count` number of filemarks.

### STFSF

Space forward the `st_count` number of filemarks.

### STRSF

Space backward the `st_count` number of filemarks.

### STFSR

Space forward the `st_count` number of records.

### STRSR

Space backward the `st_count` number of records.

### STTUR

Issue the **Test Unit Ready** command. The `st_count` parameter does not apply.

### STLOAD

Issue the **SCSI Load** command. The `st_count` parameter does not apply. The operation of the **SCSI Load** command varies depending on the type of device. See the appropriate hardware reference manual.

### STSEOD

Space forward to the end of the data. The `st_count` parameter does not apply.

### STEJECT

Unload the tape. The `st_count` parameter does not apply.

### STINSRT

Issue the **SCSI Load** command. The `st_count` parameter does not apply. The operation of the **SCSI Load** command varies depending on the type of device. See the appropriate hardware reference manual.

Note: If zero is used for operations that require the `st_count` parameter, then the command is not issued to the device. The device driver returns a successful completion. An example of the **STIOCTOP** command is

```
#include <sys/IBM_tape.h>

struct stop stop;
stop.st_op=STWEOF;
stop.st_count=3;
if (ioctl(tapefd,STIOCTOP,&stop)) {
 printf("ioctl failure. errno=%d",errno);
 exit(errno);
}
```

---

## STIOCQRYP or STIOCSETP

The **STIOCQRYP** command allows the program to query the tape device, device driver, and the media parameters. The **STIOCSETP** command allows the program to change the tape device, the device driver, and the media parameters.

Before the **STIOCSETP** command is issued, use the **STIOCQRYP** command to query and fill the fields of the data structure you do not want to change. Then, issue the **STIOCSETP** command to change the selected fields. Changing certain fields, such as `buffered_mode`, impacts performance. If the `buffered_mode` field is `FALSE`, each record that is written to the tape is immediately transferred to the tape. This operation guarantees that each record is on the tape, but it impacts performance.

## Unchangeable parameters

---

The following parameters that are returned by the **STIOCQRYP** command cannot be changed by the **STIOCSETP** command.

`hkwrdr`

This parameter is accepted but ignored.

`logical_write_protect`

This parameter sets the type of logical write protection for the tape that is loaded in the drive.

`write_protect`

If the currently mounted tape is write protected, this field is set to `TRUE`. Otherwise, it is set to `FALSE`.

`min_blksize`

This parameter is the minimum block size for the device. The driver gets this field by issuing the **SCSI Read Block Limits** command to the device.

`max_blksize`

This parameter is the maximum block size for the device. The driver gets this field by issuing the **SCSI Read Block Limits** command to the device.

`retain_reservation`

This parameter is accepted but ignored.

`medium_type`

This parameter is the media type of the currently loaded tape. Some tape devices support multiple media types and report different values in this field. See the hardware reference guide for the specific tape device to determine the possible values.

`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. Capacity Scaling is not currently supported in IBMtape.

`density_code`

This parameter is the density setting for the currently loaded tape. Some tape devices support multiple densities and report the current setting in this field. See the hardware reference guide for the specific tape device to determine the possible values.

`volid`

This field is always set to zero.

`emulate_autoloader`

This parameter is accepted but ignored.

`record_space_mode`

Only `SCSI_SPACE_MODE` is supported.

`read_sili_bit`

This parameter is accepted but ignored. SILI bit is not supported due to Linux® system environment limitations.

## Changeable parameters

---

The following parameters can be changed by using the **STIOCSETP** IOCTL command.

`trace`

This parameter turns the trace for the tape device On or Off.

`blksize`

This parameter specifies the new effective block size for the tape device. Use 0 for variable block mode.

`compression`

This parameter turns the hardware compression On or Off.

`max_scsi_xfer`

This parameter is the maximum transfer size that is allowed per SCSI command. In the IBMtape driver 3.0.3 or lower level, this value is 256 KB (262144 bytes) by default and changeable through the **STIOCSETP** IOCTL. In the IBMtape driver 3.0.5 or above and the open source driver `lin_tape`, this parameter is not changeable any more. It is determined by the maximum transfer size of the Host Bus Adapter that the tape drive is attached to.

`trailer_labels`

If this parameter is set to On, then writing a record past the early warning mark on the tape is allowed. Only the first write operation that detects the early warning mark returns the `ENOSPC` error code. All subsequent write operations are allowed to continue despite the check conditions that result from writing in the early warning zone (which are suppressed). When the end of the physical volume is reached, `EIO` is returned.

If this parameter is set to Off, the first write in the early warning zone fails, the `ENOSPC` error code is returned, and subsequent write operations fail.

`rewind_immediate`

This parameter turns the immediate bit On or Off for subsequent rewind commands. If it is set to On, then the `STREW` tape operation runs faster. However, the next tape command can take longer to finish because the actual physical rewind operation must complete before the next tape command can start.

`logging`

This parameter turns the volume logging for the tape device On or Off.

`disable_sim_logging`

If this parameter is Off, the SIM/MIM data is automatically retrieved by the IBMtape device driver whenever it is available in the tape device.

`disable_auto_drive_dump`

If this parameter is Off, the drive dump is automatically retrieved by the IBMtape device driver whenever a drive dump is in the tape device. It can also be set for all devices at modprobe configuration by adding `disable_auto_drive_dump=1`.

`logical_write_protect`

This parameter sets the type of logical write protection for the tape that is loaded in the drive. See the hardware reference guide for the specific device for different types of logical write protect.

`capacity_scaling`

This field can be changed only when the tape is positioned at the beginning of the tape. When a change is accepted, IBMtape rescales the tape capacity by formatting the loaded tape. See the *IBM Enterprise Tape System 3592 SCSI Reference* for the specific device for different types of capacity scaling. IBM® 3592 tape cartridges have two formats available, the 300 GB format and the 60 GB Fast Access format. The format of a cartridge can be queried under program control by issuing the **STIOCQRYP** IOCTL and checking the returned value of `capacity_scaling_value` (in hex).

If the `capacity_scaling_value` is 0x00, your 3592 tape cartridge is in 300 GB format. If the `capacity_scaling_value` is 0x35, your tape cartridge is in 60 GB Fast Access format. If the `capacity_scaling_value` is some other value, your tape cartridge format is undefined. (IBM can later define other supported cartridge formats.

If so, they are documented in later versions of the *IBM TotalStorage™ Enterprise Tape System 3592 SCSI Reference*).

If you want to change your cartridge format, you can use the **STIOCSETP** IOCTL to change the capacity scaling value of your cartridge.

Warning: All data on the cartridge is lost when the format is changed.

If you want to set it to the 300 GB format, set `capacity_scaling_value` to 0x00 and `capacity_scaling` to `SCALE_VALUE`. If you want to set it to the 60 GB Fast Access format, set `capacity_scaling_value` to 0x35 and `capacity_scaling` to `SCALE_VALUE`. Setting `capacity_scaling` to `SCALE_VALUE` is required.

Note: All data on the tape is deleted and is not recoverable.

`read_past_file_mark`

This parameter changes the behavior of the read function when a filemark is encountered. If the `read_past_filemark` flag is TRUE when a read operation encounters a file mark, IBMtape returns the number of bytes read before the filemark is encountered and sets the tape position at the EOT side of the file mark.

If the `read_past_filemark` flag is FALSE (by default) when a read operation encounters a filemark, if data was read, the read function returns the number of bytes read, and positions the tape at the BOT side of the filemark. If no data was read, the read returns 0 bytes and positions the tape at the EOT side of the filemark.

`limit_read_recov`

If this flag is TRUE, automatic recovery from read errors is limited to 5 seconds. If it is FALSE, the default is restored and the tape drive takes an arbitrary amount of time for read error recovery.

`limit_write_recov`

If this flag is TRUE, automatic recovery from write errors is limited to 5 seconds. If it is FALSE, the default is restored and the tape drive takes an arbitrary amount of time for write error recovery.

`data_safe_mode`

If this flag is TRUE, `data_safe_mode` is set in the drive. This action prevents data on the tape from being overwritten to avoid accidental data loss. If the value is FALSE, `data_safe_mode` is turned off.

`pews`

This parameter establishes the programmable early warning zone size. It is a 2-byte numerical value that specifies how many MB before the standard end-of-medium early warning zone to place the programmable early warning indicator. If this value is set to a positive integer, a user application is warned that the tape is running out of space when the tape head reaches the PEW location. If `pews` is set to 0, then there no early warning zone occurs and the user is notified only at the standard early warning location.

The input or output data structure is

```
struct stchgp_s {
 int blksize; /* new block size */
 boolean trace; /* TRUE = message trace on */
 uint hkwr; /* trace hook word */
 int sync_count; /* obsolete - not used */
 boolean autoloader; /* on/off autoloader feature */
 boolean buffered_mode; /* on/off buffered mode */
 boolean compression; /* on/off compression */
 boolean trailer_labels; /* on/off allow writing after EOM */
 boolean rewind_immediate; /* on/off immediate rewinds */
 boolean bus_domination; /* obsolete - not used */
 boolean logging; /* enable or disable volume logging */
 boolean write_protect; /* write_protected media */
 uint min_blksize; /* minimum block size */
 uint max_blksize; /* maximum block size */
 uint max_scsi_xfer; /* maximum scsi transfer len */
 char volid[16]; /* volume id */
 uchar acf_mode; /* automatic cartridge facility mode*/

#define ACF_NONE 0
#define ACF_MANUAL 1
#define ACF_SYSTEM 2
#define ACF_AUTOMATIC 3
#define ACF_ACCUMULATE 4
#define ACF_RANDOM 5
 uchar record_space_mode; /* fsr/bsr space mode */

#define SCSI_SPACE_MODE 1
#define AIX_SPACE_MODE 2
 uchar logical_write_protect; /* logical write protect */

#define NO_PROTECT 0
#define ASSOCIATED_PROTECT 1
#define PERSISTENT_PROTECT 2
#define WORM_PROTECT 3
 uchar capacity_scaling; /* capacity scaling */

#define SCALE_100 0
#define SCALE_75 1
#define SCALE_50 2
#define SCALE_25 3
#define SCALE_VALUE 4
 uchar retain_reservation; /* retain reservation */
 uchar alt_pathing; /* alternate pathing active */
 boolean emulate_autoloader; /* emulate autoloader in random mode*/
 uchar medium_type; /* tape medium type */
 uchar density_code; /* tape density code */
 boolean disable_sim_logging; /* disable sim/mim error logging */
 boolean read_sili_bit; /* SILI bit setting for read commands*/
 uchar read_past_filemark; /* fixed block read pass the filemark*/
 boolean disable_auto_drive_dump; /* disable auto drive dump logging*/
 uchar capacity_scaling_value; /* hex value of capacity scaling */
 boolean wfm_immediate; /* buffer write file mark */
 boolean limit_read_recov; /* limit read recovery to 5 seconds */
 boolean limit_write_recov; /* limit write recovery to 5 seconds*/
 boolean data_safe_mode; /* turn data safe mode on/off */
 uchar pews[2]; /* programmable early warn zone size*/
 uchar reserve_type; /* if set persistent reserve will be used */
 uchar reserved[12];
};
```

An example of the **STIOCQRP** and **STIOCSETP** commands is

```
#include <sys/IBM_tape.h>
struct stchgp_s stchgp;
```

```

/* get current parameters */
if (ioctl(tapefd, STIOCQRYP, &stchgp)) {
 printf("ioctl failure. errno=%d", errno);
 exit(errno);
}
/* set new parameters */
stchgp.rewind_immediate=1;
stchgp.trailer_labels=1;
if (ioctl(tapefd, STIOCSETP, &stchgp)) {
 printf("IOCTL failure. errno=%d", errno);
 exit(errno);
}

```

---

## STIOCSYNC

This IOCTL command immediately flushes the tape buffers to the tape. There are no arguments for this IOCTL command.

An example of the **STIOCSYNC** command is

```

#include <sys/IBM_tape.h>
if (ioctl(tapefd, STIOCSYNC, NULL)) {
 printf("ioctl failure. errno=%d", errno);
 exit(errno);
}

```

---

## STIOCDM

This IOCTL command shows and manipulates one or two messages on the message display. The message that is sent with this call does not always remain on the display. It depends on the current state of the tape device. Refer to the IBM® 3590 manuals for a description of the message display functions.

The input data structure is

```

#define MAXMSGLEN 8
struct stdm_s
{
 char dm_function; /* function code */
 /* function selection */
 #define DMSTATUSMSG 0x00 /* general status message */
 #define DMDVMSG 0x20 /* demount verify message */
 #define DMMIMMED 0x40 /* mount with immediate action indicator */
 #define DMDMIMMED 0xE0 /* demount/mount with immediate action */
 /* message control */
 #define DMMSG0 0x00 /* display message 0 */
 #define DMMSG1 0x04 /* display message 1 */
 #define DMFLASHMSG0 0x08 /* flash message 0 */
 #define DMFLASHMSG1 0x0C /* flash message 1 */
 #define DMALTERNATE 0x10 /* alternate message 0 and message 1 */
 char dm_msg0[MAXMSGLEN]; /* message 0 */
 char dm_msg1[MAXMSGLEN]; /* message 1 */
};

```

An example of the **STIOCDM** command is

```

#include <sys/IBM_tape.h>
struct stdm_s stdm;
memset(&stdm, 0, sizeof(struct stdm_s));
stdm.dm_func = DMSTATUSMSG|DMMSG0;
bcopy("SSG", stdm.dm_msg0, 8);
if (ioctl(tapefd, STIOCDM, &stdm) < 0)
{
 printf("IOCTL failure, errno = %d", errno);
 exit(errno);
}

```

---

## STIOCQRYPOS

This command queries the tape position. Tape position is defined as the location where the next read or write operation occurs. The query function can be used independently of, or with, the **STIOCSETPOS** IOCTL command.

A write filemark of count 0 is always issued to the drive, which flushes all data from the buffers to the tape media. After the write filemark finishes, the query is issued.

After a query operation, the **curpos** field is set to an unsigned integer that represents the current position.

The **eot** field is set to TRUE if the tape is positioned between the early warning and the physical end of the tape. Otherwise, it is set to FALSE.

The **lbot** field is valid only if the last command was a **write** command. If a query is issued and the last command was not a write, **lbot** contains the value **LBOT\_UNKNOWN**.

Note: **lbot** indicates the last block of data that is transferred to the tape.

The number of blocks and number of bytes currently in the tape device buffers is returned in the **num\_blocks** and **num\_bytes** fields.

The **bot** field is set to TRUE if the tape position is at the beginning of the tape. Otherwise, it is set to FALSE.

The returned partition\_number field is the current partition of the loaded tape.

The block ID of the next block of data to be transferred to or from the physical tape is returned in the tapepos field.

The position data structure is

```
typedef unsigned int blockid_t;
struct stpos_s {
 char _block_type; /* Format of block ID information */
 #define QP_LOGICAL 0 /* SCSI logical block ID format */
 #define QP_PHYSICAL 1 /* Vendor-specific block ID format */
 boolean eot; /* Position is after early warning, */
 /* before physical end of tape. */
 blockid_t curpos; /* For query pos, current position. */
 /* For set pos, position to go to. */
 blockid_t lbot; /* Last block written to tape. */
 #define LBOT_NONE 0xFFFFFFFF /* No blocks written to tape. */
 #define LBOT_UNKNOWN 0xFFFFFEE /* Unable to determine info. */
 uint num_blocks; /* Number of blocks in buffer. */
 uint num_bytes; /* Number of bytes in buffer. */
 boolean bot; /* Position is at beginning of tape */
 uchar partition_number; /* Current partition number on tape */
 uchar reserved1[2];
 blockid_t tapepos; /* Next block to be transferred. */
 uchar reserved2[48];
};
```

An example of the **STIOCQRYPOS** command is

```
#include <sys/IBM_tape.h>
struct stpos_s stpos;
stpos.block_type=QP_PHYSICAL;
if (ioctl(tapefd,STIOCQRYPOS,&stpos)) {
 printf("ioctl failure. errno=%d",errno);
 exit(errno);
}
oldposition=stpos.curpos;
```

---

## STIOCSETPOS

This IOCTL command issues a high speed locate operation to the position specified on the tape. It uses the same position data structure that is described for **STIOCQRYPOS**, however, only the block\_type and curpos fields are used during a set operation. **STIOCSETPOS** can be used independently of or with **STIOCQRYPOS**.

The block\_type must be set to either QP\_PHYSICAL or QP\_LOGICAL. However, there is no difference in how IBMtape processes the request.

An example of the **STIOCQRYPOS** and **STIOCSETPOS** commands is

```
#include <sys/IBM_tape.h>
struct stpos_s stpos;
stpos.block_type=QP_LOGICAL;
if (ioctl(tapefd,STIOCQRYPOS,&stpos)) {
 printf("ioctl failure. errno=%d",errno);
 exit(errno);
}
oldposition=stpos.curpos;

stpos.curpos=oldposition;
stpos.block_type=QP_LOGICAL;
if (ioctl(tapefd,STIOCSETPOS,&stpos)) {
 printf("ioctl failure. errno=%d",errno);
 exit(errno);
}
```

---

## STIOCQRYSENSE

This IOCTL command returns the last sense data that is collected from the tape device. Or, it issues a new **Request Sense** command and returns the collected data. If sense\_type equals LASTERROR, then the sense data is valid only if the last tape operation had an error that caused a sense command to be issued to the device. If the sense data is valid, then the IOCTL command finishes successfully, and the len field is set to a value greater than zero. The residual\_count field contains the residual count from the last operation.

The input or output data structure is

```
#define MAXSENSE 255
struct stsense_s {
 /* input */
 char sense_type; /* fresh (new sense) or sense from last error */
 #define FRESH 1 /* Initiate a new sense command */
 #define LASTERROR 2 /* Return sense gathered from */
 /* the last SCSI sense command. */

 /* output */
 uchar sense[MAXSENSE]; /* actual sense data */
 int len; /* length of valid sense data returned */
 int residual_count; /* residual count from last operation */
 uchar reserved[60];
};
```

An example of the **STIOCQRYSENSE** command is

```
#include <sys/IBM_tape.h>
struct stsense_s stsense;
stsense.sense_type=LASTERROR;
#define MEDIUM_ERROR 0x03
if (ioctl(tapefd,STIOCQRYSENSE,&stsense)) {
 printf("ioctl failure. errno=%d",errno);
 exit(errno);
}
if ((stsense.sense[2]&0x0F)==MEDIUM_ERROR) {
 printf("We're in trouble now!");
 exit(SENSE_KEY(&stsense.sense));
}
```

---

## STIOCQRYINQUIRY

This IOCTL command returns the inquiry data from the device. The data is divided into standard and vendor-specific portions.

The output data structure is

```
/*inquiry data info */
struct inq_data_s {
 BYTE b0;
 /*macros for accessing fields of byte 1 */
 #define PERIPHERAL_QUALIFIER(x) ((x->b0 &0xE0)>>5)
 #define PERIPHERAL_CONNECTED 0x00
 #define PERIPHERAL_NOT_CONNECTED 0x01
 #define LUN_NOT_SUPPORTED 0x03
 #define PERIPHERAL_DEVICE_TYPE(x) (x->b0 &0x1F)
 #define DIRECT_ACCESS 0x00
 #define SEQUENTIAL_DEVICE 0x01
 #define PRINTER_DEVICE 0x02
 #define PROCESSOR_DEVICE 0x03
 #define CD_ROM_DEVICE 0x05
 #define OPTICAL_MEMORY_DEVICE 0x07
 #define MEDIUM_CHANGER_DEVICE 0x08
 #define UNKNOWN 0x1F
 BYTE b1;
 /*macros for accessing fields of byte 2 */
 #define RMB(x) ((x->b1 &0x80)>>7) /*removable media bit */
 #define FIXED 0
 #define REMOVABLE 1
 #define device_type_qualifier(x) (x->b1 &0x7F) /*vendor specific */
 BYTE b2;
 /*macros for accessing fields of byte 3 */
 #define ISO_Version(x) ((x->b2 &0xC0)>>6)
 #define ECMA_Version(x) ((x->b2 &0x38)>>3)
 #define ANSI_Version(x) (x->b2 &0x07)
 #define NONSTANDARD 0
 #define SCSI1 1
 #define SCSI2 2
 #define SCSI3 3
 BYTE b3;
 /*macros for accessing fields of byte 4 */
 /* asynchronous event notification */
 #define AENC(x) ((x->b3 &0x80)>>7)
 /* support terminate I/O process message? */
 #define TrmIOP(x) ((x->b3 &0x40)>>6)
 #define Response_Data_Format(x) (x->b3 &0x0F)
 #define SCSI1INQ 0 /* SCSI-1 standard inquiry data format */
 #define CCSINQ 1 /* CCS standard inquiry data format */
 #define SCSI2INQ 2 /* SCSI-2 standard inquiry data format */
 BYTE additional_length; /* bytes following this field minus 4 */
 BYTE res5;
 BYTE b6;
 #define MChngr(x) ((x->b6 & 0x08)>>3)
 BYTE b7;
 /*macros for accessing fields of byte 7 */
 #define RelAdr(x) ((x->b7 &0x80)>>7)
 /* the following fields are true or false */
 #define WBus32(x) ((x->b7 &0x40)>>6)
 #define WBus16(x) ((x->b7 &0x20)>>5)
 #define Sync(x) ((x->b7 &0x10)>>4)
 #define Linked(x) ((x->b7 &0x08)>>3)
 #define CmdQue(x) ((x->b7 &0x02)>>1)
 #define SftRe(x) (x->b7 &0x01)
 char vendor_identification [8];
 char product_identification [16];
 char product_revision_level [4];
};
struct st_inquiry
{
 struct inq_data_s standard;
 BYTE vendor_specific [255-sizeof(struct inq_data_s)];
};
```

An example of the **STIOCQRYINQUIRY** command is

```
struct st_inquiry inqd;
if (ioctl(tapefd,STIOCQRYINQUIRY,&inqd)) {
```



```

printf("ioctl failure. errno=%d\n",errno);
exit(errno);
}
if (ANSI_Version(((struct inq_data s *)&(inqd.standard)))==SCSI2)
printf("Hey! We have a SCSI-2 device\n");

```

---

## STIOC\_LOCATE

This IOCTL command causes the tape to be positioned at the specified block ID. The block ID used for the command must be obtained by using the **STIOC\_READ\_POSITION** command.

An example of the **STIOC\_LOCATE** command is

```

#include <sys/IBM_tape.h>
unsigned int current_blockid;

/* read current tape position */
if (ioctl(tapefd,STIOC_READ_POSITION,&current_blockid)) {
 printf("ioctl failure. errno=%d\n",errno);
 exit(1);
}

/* restore current tape position */
if (ioctl(tapefd,STIOC_LOCATE,current_blockid)) {
 printf("ioctl failure. errno=%d\n",errno);
 exit(1);
}

```

---

## STIOC\_READ\_POSITION

This IOCTL command returns the block ID of the current position of the tape. The block ID returned from this command can be used with the **STIOC\_LOCATE** command to set the position of the tape.

An example of the **STIOC\_READ\_POSITION** command is

```

#include <sys/IBM_tape.h>
unsigned int current_blockid;
/* read current tape position */
if (ioctl(tapefd,STIOC_READ_POSITION,&current_blockid)) {
 printf("ioctl failure. errno=%d\n",errno);
 exit(1);
}
/* restore current tape position */
if (ioctl(tapefd,STIOC_LOCATE,current_blockid)) {
 printf("ioctl failure. errno=%d\n",errno);
 exit(1);
}

```

---

## STIOC\_RESET\_DRIVE

This IOCTL command issues a **SCSI Send Diagnostic** command to reset the tape drive. There are no arguments for this IOCTL command.

An example of the **STIOC\_RESET\_DRIVE** command is

```

/* reset the tape drive */
if (ioctl(tapefd,STIOC_RESET_DRIVE,NULL)) {
 printf("ioctl failure. errno=%d\n",errno);
 exit(errno);
}

```

---

## STIOC\_PREVENT\_MEDIUM\_REMOVAL

This IOCTL command prevents an operator from removing media from the device until the **STIOC\_ALLOW\_MEDIUM\_REMOVAL** command is issued or the device is reset.

There is no associated data structure.

An example of the **STIOC\_PREVENT\_MEDIUM\_REMOVAL** command is

```

#include <sys/IBM_tape.h>
if (!ioctl (tapefd, STIOC_PREVENT_MEDIUM_REMOVAL, NULL))
 printf ("The STIOC_PREVENT_MEDIUM_REMOVAL ioctl succeeded\n");
else {
 perror ("The STIOC_PREVENT_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}

```

---

## STIOC\_ALLOW\_MEDIUM\_REMOVAL

This IOCTL command allows an operator to remove media from the device. This command is normally used after the **STIOC\_PREVENT\_MEDIUM\_REMOVAL** command to restore the device to the default state.

There is no associated data structure.

An example of the **STIOC\_ALLOW\_MEDIUM\_REMOVAL** command is

```
#include <sys/IBM_tape.h>
if (!ioctl (tapefd, STIOC_ALLOW_MEDIUM_REMOVAL, NULL))
 printf ("The STIOC_ALLOW_MEDIUM_REMOVAL ioctl succeeded\n");
else {
 perror ("The STIOC_ALLOW_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}
```

---

## STIOC\_REPORT\_DENSITY\_SUPPORT

This IOCTL command issues the **SCSI Report Density Support** command to the tape device. It returns either ALL supported densities or only supported densities for the currently mounted media. The media field specifies which type of report is requested. The number\_reports field is returned by the device driver and indicates how many density reports in the reports array field were returned.

The data structures that are used with this IOCTL is

```
struct density_report {
 unchar primary_density_code; /* primary density code */
 unchar secondary_density_code; /* secondary density code */
 uint wrtok :1, /* write ok, device can write this format */
 dup :1, /* zero if density only reported once */
 deflt :1, /* current density is default format */
 :5; /* reserved */
 char reserved[2]; /* reserved */
 uint bits_per_mm :24; /* bits per mm */
 ushort media_width; /* media width in millimeters */
 ushort tracks; /* tracks */
 uint capacity; /* capacity in megabytes */
 char assigning_org[8]; /* assigning organization in ASCII */
 char density_name[8]; /* density name in ASCII */
 char description[20]; /* description in ASCII */
};

struct report_density_support {
 unchar media; /* report all or current media as defined above */
 ushort number_reports; /* number of density reports returned in array */
 struct density_report reports[MAX_DENSITY_REPORTS];
};
```

Examples of the **STIOC\_REPORT\_DENSITY\_SUPPORT** command are

```
#include <sys/IBM_tape.h>
int stioc_report_density_support(void)
{
 int i;
 struct report_density_support density;
 printf("Issuing Report Density Support for ALL supported media...\n");
 density.media = ALL_MEDIA_DENSITY;
 if (ioctl(fdd, STIOC_REPORT_DENSITY_SUPPORT, &density) != 0)
 return errno;
 printf("Total number of densities reported:
 %d\n", density.number_reports);
 for (i = 0; i < density.number_reports; i++) {
 printf("\n");
 printf(" Density Name..... %0.8s\n",
 density.reports[i].density_name);
 printf(" Assigning Organization..... %0.8s\n",
 density.reports[i].assigning_org);
 printf(" Density Name..... %0.8s\n",
 density.reports[i].density_name);
 printf(" Description..... %0.20s\n",
 density.reports[i].description);
 printf(" Primary Density Code..... %02X\n",
 density.reports[i].primary_density_code);
 printf(" Secondary Density Code..... %02X\n",
 density.reports[i].secondary_density_code);
 if (density.reports[i].wrtok)
 printf(" Write OK..... Yes\n");
 else
 printf(" Write OK..... No\n");
 if (density.reports[i].dup)
 printf(" Duplicate..... Yes\n");
 else
 printf(" Duplicate..... No\n");
 if (density.reports[i].deflt)
 printf(" Default..... Yes\n");
 else
 printf(" Default..... No\n");
 printf(" Bits per MM..... %d\n",
```

```

 density.reports[i].bits_per_mm);
 printf(" Media Width (millimeters).... %d\n",
 density.reports[i].media_width);
 printf(" Tracks..... %d\n",
 density.reports[i].tracks);
 printf(" Capacity (megabytes)..... %d\n",
 density.reports[i].capacity);
 if (opcode) {
 printf ("\\nHit enter> to continue?");
 getchar();
 }
}
printf("\\nIssuing Report Density Support for CURRENT media...\\n");
density.media = CURRENT_MEDIA_DENSITY;
if (ioctl(fd, STIOC_REPORT_DENSITY_SUPPORT, &density) != 0)
 return errno;
for (i = 0; i < density.number_reports; i++) {
 printf("\\n");
 printf(" Density Name..... %0.8s\\n",
 density.reports[i].density_name);
 printf(" Assigning Organization..... %0.8s\\n",
 density.reports[i].assigning_org);
 printf(" Description..... %0.20s\\n",
 density.reports[i].description);
 printf(" Primary Density Code..... %02X\\n",
 density.reports[i].primary_density_code);
 printf(" Secondary Density Code..... %02X\\n",
 density.reports[i].secondary_density_code);
 if (density.reports[i].wrtok)
 printf(" Write OK..... Yes\\n");
 else
 printf(" Write OK..... No\\n");
 if (density.reports[i].dup)
 printf(" Duplicate..... Yes\\n");
 else
 printf(" Duplicate..... No\\n");
 if (density.reports[i].deflt)
 printf(" Default..... Yes\\n");
 else
 printf(" Default..... No\\n");
 printf(" Bits per MM..... %d\\n",
 density.reports[i].bits_per_mm);
 printf(" Media Width (millimeters).... %d\\n",
 density.reports[i].media_width);
 printf(" Tracks..... %d\\n",
 density.reports[i].tracks);
 printf(" Capacity (megabytes)..... %d\\n",
 density.reports[i].capacity);
}
return errno;
}

```

---

## MTDEVICE (Obtain Device Number)

This IOCTL command obtains the device number that is used for communicating with a 3494 library.

An example of the **MTDEVICE** command is

```

int device;
if(ioctl(tapefd, MTDEVICE, &device)<0)
{
 printf("IOCTL failure, errno = %d\\n", errno);
 exit(errno);
}
printf("Device number is %X\\n", device);

```

---

## STIOC\_GET\_DENSITY and STIOC\_SET\_DENSITY

The **STIOC\_GET\_DENSITY** IOCTL is used to query the current write density format settings on the tape drive. The current density code from the drive Mode Sense header, the Read/Write Control Mode page default density and pending density are returned.

The **STIOC\_SET\_DENSITY** IOCTL is used to set a new write density format on the tape drive by using the default and pending density fields. The density code field is not used and ignored on this IOCTL. The application can specify a new write density for the current loaded tape only or as a default for all tapes. Refer to the examples below.

The application must get the current density settings first before the current settings are modified. If the application specifies a new density for the current loaded tape only, then the application must issue another set density IOCTL after the current tape is unloaded and the next tape is loaded to either the default maximum density or a new density to ensure the tape drive uses the correct density. If the application specifies a new default density for all tapes, the setting remains in effect until changed by another set density IOCTL or the tape drive is closed by the application.

The structure for the **STIOC\_GET\_DENSITY** and **STIOC\_SET\_DENSITY** IOCTLs is

```

struct density_data_t
{
 char density_code; /* mode sense header density code */
 char default_density; /* default write density */
 char pending_density; /* pending write density */
}

```

```
char reserved[9];
};
```

Note:

1. These IOCTLs are supported only on tape drives that can write multiple density formats. Refer to the Hardware Reference for the specific tape drive to determine whether multiple write densities are supported. If the tape drive does not support these IOCTLs, *errno* EINVAL is returned.
2. The device driver always sets the default maximum write density for the tape drive on every open system call. Any previous **STIOC\_SET\_DENSITY** IOCTL values from the last open are not used.
3. If the tape drive detects an invalid density code or cannot run the operation on the **STIOC\_SET\_DENSITY** IOCTL, the *errno* is returned and the current drive density settings before the IOCTL are restored.
4. The struct `density_data_t` defined in the header file is used for both IOCTLs. The `density_code` field is not used and ignored on the **STIOC\_SET\_DENSITY** IOCTL.

## Examples

```
struct density_data_t data;

/* open the tape drive */
/* get current density settings */
rc = ioctl(fd, STIOC_GET_DENSITY, &data);

/* set 3592 J1A density format for current loaded tape only */
data.default_density = 0x7F;
data.pending_density = 0x51;
rc = ioctl(fd, STIOC_SET_DENSITY, &data);

/* unload tape */
/* load next tape */
/* set 3592 E05 density format for current loaded tape only */
data.default_density = 0x7F;
data.pending_density = 0x52;
rc = ioctl(fd, STIOC_SET_DENSITY, &data);

/* unload tape */
/* load next tape */
/* set default maximum density for current loaded tape */
data.default_density = 0;
data.pending_density = 0;
rc = ioctl(fd, STIOC_SET_DENSITY, &data);

/* close the tape drive */
/* open the tape drive */
/* set 3592 J1A density format for current loaded tape and all subsequent tapes */
data.default_density = 0x51;
data.pending_density = 0x51;

rc = ioctl(fd, STIOC_SET_DENSITY, &data);
```

## GET\_ENCRYPTION\_STATE

This IOCTL command queries the drive's encryption method and state. The data structure that is used for this IOCTL is as follows on all of the supported operating systems

```
struct encryption_status
{
 uchar encryption_capable; /* (1) Set this field as a boolean based on the
 capability of the drive */
 uchar encryption_method; /* (2) Set this field to one of the following */
 #define METHOD_NONE 0 /* Only used in GET_ENCRYPTION_STATE */
 #define METHOD_LIBRARY 1 /* Only used in GET_ENCRYPTION_STATE */
 #define METHOD_SYSTEM 2 /* Only used in GET_ENCRYPTION_STATE */
 #define METHOD_APPLICATION 3 /* Only used in GET_ENCRYPTION_STATE */
 #define METHOD_CUSTOM 4 /* Only used in GET_ENCRYPTION_STATE */
 #define METHOD_UNKNOWN 5 /* Only used in GET_ENCRYPTION_STATE */
 uchar encryption_state; /* (3) Set this field to one of the following */
 #define STATE_OFF 0 /* Used in GET/SET_ENCRYPTION_STATE */
 #define STATE_ON 1 /* Used in GET/SET_ENCRYPTION_STATE */
 #define STATE_NA 2 /* Only used in GET_ENCRYPTION_STATE */
 uchar[13] reserved;
};
```

An example of the **GET\_ENCRYPTION\_STATE** command is

```
int qry_encryption_state (void)
{
 int rc = 0;
 struct encryption_status encryption_status_t;

 printf("issuing query encryption status...\n");
 memset(&encryption_status_t, 0, sizeof(struct encryption_status));
 rc = ioctl(fd, GET_ENCRYPTION_STATE, &encryption_status_t);
 if(rc == 0)
 {
 if(encryption_status_t.encryption_capable)
 printf("encryption capable.....Yes\n");
 else
 printf("encryption capable.....No\n");
 switch(encryption_status_t.encryption_method)
```

```

{
case METHOD_NONE:
 printf("encryption method.....METHOD_NONE\n");
 break;
case METHOD_LIBRARY:
 printf("encryption method.....METHOD_LIBRARY\n");
 break;
case METHOD_SYSTEM:
 printf("encryption method.....METHOD_SYSTEM\n");
 break;
case METHOD_APPLICATION:
 printf("encryption method.....METHOD_APPLICATION\n");
 break;
case METHOD_CUSTOM:
 printf("encyrpton method.....METHOD_CUSTOM\n");
 break;
case METHOD_UNKNOWN:
 printf("encryption method.....METHOD_UNKNOWN\n");
 break;

default:
 printf("encryption method.....Error\n");
}

switch(encryption_status_t.encryption_state)
{
case STATE_OFF:
 printf("encryption state.....OFF\n");
 break;
case STATE_ON:
 printf("encryption state.....ON\n");
 break;
case STATE_NA:
 printf("encryption state.....NA\n");
 break;

default:
 printf("encryption state.....Error\n");
}
}

return rc;
}

```

---

## SET\_ENCRYPTION\_STATE

This IOCTL command allows setting the encryption state only for application-managed encryption. On unload, some drive settings might be reset to default. To set the encryption state, the application must issue this IOCTL after a tape is loaded and at BOP.

The data structure that is used for this IOCTL is the same as the one for **GET\_ENCRYPTION\_STATE**. An example of the **SET\_ENCRYPTION\_STATE** command is

```

int set_encryption_state(int option)
{
 int rc = 0;
 struct encryption_status encryption_status_t;

 printf("issuing query encryption status...\n");
 memset(&encryption_status_t, 0, sizeof(struct encryption_status));
 rc = ioctl(fd, GET_ENCRYPTION_STATE, &encryption_status_t);
 if(rc < 0) return rc;

 if(option == 0)
 encryption_status_t.encryption_state = STATE_OFF;
 else if(option == 1)
 encryption_status_t.encryption_state = STATE_ON;
 else
 {
 printf("Invalid parameter.\n");
 return -EINVAL;
 }

 printf("Issuing set encryption state.....\n");
 rc = ioctl(fd, SET_ENCRYPTION_STATE, &encryption_status_t);

 return rc;
}

```

---

## SET\_DATA\_KEY

This IOCTL command allows the data key to be set only for application-managed encryption. The data structure that is used for this IOCTL is as follows on all of the supported operating systems.

```

struct data_key
{
 uchar[12] data_key_index;
 uchar data_key_index_length;
 uchar[15] reserved1;
}

```

```

 uchar[32] data_key;
 uchar[48] reserved2;
};

```

An example of the **SET\_DATA\_KEY** command is

```

int set_datakey(void)
{
 int rc = 0;
 struct data_key encryption_data_key_t;

 printf("Issuing set encryption data key.....\n");
 memset(&encryption_data_key_t 0, sizeof(struct data_key));
 /* fill in your data key here, then issue the following ioctl*/
 rc = ioctl(fd, SET_DATA_KEY, &encryption_data_key_t);
 return rc;
}

```

---

## STIOC\_QUERY\_PARTITION

This IOCTL queries and displays information for tapes that support partitioning. The data structure that is used for this IOCTL is

```

#define MAX_PARTITIONS 255
struct query_partition {
 uchar max_partitions;
 uchar active_partition;
 uchar number_of_partitions;
 uchar size_unit;
 ushort size[MAX_PARTITIONS];
 char reserved[32];
};

```

- max\_partitions is the maximum number of partitions that the tape allows.
- active\_partition is the current partition to which tape operations apply.
- number\_of\_partitions is the number of partitions currently on the tape.
- size\_unit describes the units for the size of the tape, which is given as a logarithm to the base 10.

For example, 0 refers to  $10^0 = 1$ , the most basic unit, which is bytes. All sizes that are reported are in bytes. 3 refers to  $10^3$ , or kilobytes. Size is an array of the size of the partitions on tape, one array element per partition, in size\_units.

An example of the **STIOC\_QUERY\_PARTITION** IOCTL is

```

int stioc_query_partition()
{
 struct query_partition qry;
 int rc = 0, i = 0;

 memset(&qry, '\0', sizeof(struct query_partition));
 printf("Issuing IOCTL...\n");
 rc = ioctl(fd, STIOC_QUERY_PARTITION, &qry);

 if(rc) {
 printf("Query partition failed: %d\n", rc);
 goto EXIT_LABEL;
 } /* if */

 printf("\nmax possible partitions: %d\n", qry.max_partitions);
 printf("number currently on tape: %d\n", qry.number_of_partitions);
 printf("active: %d\n", qry.active_partition);
 printf("unit: %d\n", qry.size_unit);

 for(i = 0; i < qry.number_of_partitions; i++)
 printf("size[%d]: %d\n", i, qry.size[i]);

EXIT_LABEL:

 return rc;
} /* stioc_query_partition() */

```

---

## STIOC\_CREATE\_PARTITION

This IOCTL creates partitions on tapes that support partitioning. The data structure that is used for this IOCTL is

```

#define IDP_PARTITION (1)
#define SDP_PARTITION (2)
#define FDP_PARTITION (3)
struct tape_partition {
 uchar type;
 uchar number_of_partitions;
 uchar size_unit;
 ushort size[MAX_PARTITIONS];
 char reserved[32];
};

```

Type is the type of partition, whether IDP\_PARTITION (initiator defined partition), SDP\_PARTITION (select data partition), or FDP\_PARTITION (fixed data partition). The behavior of these options is described in the SCSI reference for your tape drive.

- number\_of\_partitions is the number of partitions the user wants to create.
- size\_unit is as defined in the STIOC\_QUERY\_PARTITION section.
- size is an array of requested sizes, in size\_units, one array element per partition.

An example of the **STIOC\_CREATE\_PARTITION** IOCTL is

```
int stioc_create_partition()
{
 int rc = 0, i = 0, char_cap = 0, short_cap = 0;
 struct tape_partition crt;
 char* input = NULL;

 char_cap = pow(2, sizeof(char) * BITS_PER_BYTE) - 1;
 short_cap = pow(2, sizeof(short) * BITS_PER_BYTE) - 1;

 input = malloc(DEF_BUF_SIZE / 16);
 if(!input) {
 rc = ENOMEM;
 goto EXIT_LABEL;
 } /* if */
 memset(input, '\0', DEF_BUF_SIZE / 16);

 memset(&crt, '\0', sizeof(struct tape_partition));

 while(atoi(input) < IDP_PARTITION || atoi(input) > FDP_PARTITION + 1) {
 printf("%d) IDP_PARTITION\n", IDP_PARTITION);
 printf("%d) SDP_PARTITION\n", SDP_PARTITION);
 printf("%d) FDP_PARTITION\n", FDP_PARTITION);
 printf("%d) Cancel\n", FDP_PARTITION + 1);
 printf("\nPlease select: ");

 fgets(input, DEF_BUF_SIZE / 16, stdin);
 if(atoi(input) == FDP_PARTITION + 1) {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */
 } /* while */

 crt.type = atoi(input);

 memset(input, '\0', DEF_BUF_SIZE / 16);
 while(input[0] < '1' || input[0] > '9') {
 printf("Enter desired number of partitions (0 to cancel): ");
 fgets(input, DEF_BUF_SIZE / 16, stdin);
 if(input[0] == '\0') {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */

 if(atoi(input) > MAX_PARTITIONS) {
 printf("Please select number <= %d\n", MAX_PARTITIONS);
 input[0] = '\0';
 } /* if */
 } /* while */

 crt.number_of_partitions = atoi(input);

 if(crt.type == IDP_PARTITION && crt.number_of_partitions > 1) {
 memset(input, '\0', DEF_BUF_SIZE / 16);
 while(input[0] < '0' || input[0] > '9') {
 printf("Enter size unit (0 to cancel): ");
 fgets(input, DEF_BUF_SIZE / 16, stdin);
 if(input[0] == '\0') {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */
 if(atoi(input) > char_cap) {
 printf("Please select number <= %d\n", char_cap);
 input[0] = '\0';
 } /* if */
 } /* while */
 crt.size_unit = atoi(input);

 for(i = 0; i < crt.number_of_partitions; i++) {
 memset(input, '\0', DEF_BUF_SIZE / 16);
 while(input[0] != '-' &&
 (input[0] < '0' || input[0] > '9')) {
 printf("Enter size[%d] (0 to cancel, < 0 for "\
 "remaining space on cartridge): ", i);
 fgets(input, DEF_BUF_SIZE / 16, stdin);
 if(input[0] == '\0') {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */

 if(atoi(input) > short_cap) {
 printf("Please select number <= %d\n",
 short_cap);
 input[0] = '\0';
 } /* if */
 } /* while */
 if(input[0] == '-' && atoi(&input[1]) > 0)
 crt.size[i] = 0xFFFF;
 else crt.size[i] = atoi(input);
 } /* for */
 }
}
```

```

 } /* if */

 printf("Issuing IOCTL...\n");
 rc = ioctl(fd, STIOC_CREATE_PARTITION, &crt);

 if(rc) {
 printf("Create partition failed: %d\n", rc);
 goto EXIT_LABEL;
 } /* if */

EXIT_LABEL:

 if(input) free(input);
 return rc;
} /* stioc_create_partition() */

```

---

## STIOC\_SET\_ACTIVE\_PARTITION

This IOCTL allows the user to specify the partition on which to run subsequent tape operations. The data structure that is used for this IOCTL is

```

struct set_active_partition {
 unchar partition_number;
 unsigned long long logical_block_id;
 char reserved[32];
};

```

- partition\_number is the number of the requested active partition.
- logical\_block\_id is the requested block position within the new active partition.

An example of the **STIOC\_SET\_ACTIVE\_PARTITION** IOCTL is

```

int stioc_set_partition()
{
 int rc = 0;
 struct set_active_partition set;
 char* input = NULL;

 input = malloc(DEF_BUF_SIZE / 16);
 if(!input) {
 rc = ENOMEM;
 goto EXIT_LABEL;
 } /* if */
 memset(input, '\0', DEF_BUF_SIZE / 16);

 memset(&set, '\0', sizeof(struct set_active_partition));
 while(input[0] < '0' || input[0] > '9') {
 printf("Select partition (< 0 to cancel): ");
 fgets(input, DEF_BUF_SIZE / 16, stdin);

 if(input[0] == '-' && atoi(&input[1]) > 0) {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */

 if(atoi(input) > MAX_PARTITIONS) {
 printf("Please select number < %d\n", MAX_PARTITIONS);
 input[0] = '\0';
 } /* if */
 } /* while */
 set.partition_number = atoi(input);

 printf("Issuing IOCTL...\n");
 rc = ioctl(fd, STIOC_SET_ACTIVE_PARTITION, &set);
 if(rc) {
 printf("Set partition failed: %d\n", rc);
 goto EXIT_LABEL;
 } /* if */

EXIT_LABEL:

 if(input) free(input);
 return rc;
} /* stioc_set_partition() */

```

---

## STIOC\_ALLOW\_DATA\_OVERWRITE

This IOCTL allows data on the tape to be overwritten when in data safe mode. The data structure that is used for this IOCTL is

```

struct allow_data_overwrite {
 unchar partition_number;
 unsigned long long logical_block_id;
 unchar allow_format_overwrite;
 char reserved[32];
};

```

- partition\_number is the number of the drive partition on which to allow the overwrite.
- logical\_block\_id is the block that you want to overwrite.



- `allow_format_overwrite`, if set to `TRUE`, instructs the tape drive to allow a format of the tape and accept the **CREATE\_PARTITION** ioctl.

If `allow_format_overwrite` is `TRUE`, `partition_number` and `logical_block_id` are ignored.

An example of the use of the **STIOC\_ALLOW\_DATA\_OVERWRITE** IOCTL is

```
int stioc_allow_overwrite()
{
 int rc = 0, i = 0, brk = FALSE;
 struct allow_data_overwrite ado;
 char* input = NULL;

 memset(&ado, '\0', sizeof(struct allow_data_overwrite));
 input = malloc(DEF_BUF_SIZE / 4);
 if(!input) {
 rc = ENOMEM;
 goto EXIT_LABEL;
 } /* if */
 memset(input, '\0', DEF_BUF_SIZE / 4);

 while(input[0] < '0' || input[0] > '1') {
 printf("0. Write Data 1. Create Partition (< 0 to cancel): ");
 fgets(input, DEF_BUF_SIZE / 4, stdin);

 if(input[0] == '-' && atoi(&input[1]) > 0) {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */
 } /* while */

 ado.allow_format_overwrite = atoi(&input[0]);
 switch(ado.allow_format_overwrite) {
 case 0:
 memset(input, '\0', DEF_BUF_SIZE / 4);
 while((input[0] < '0' || input[0] > '9')
 && (input[0] < 'a' || input[0] > 'f')) {
 brk = FALSE;
 printf("Enter partition in hex (< 0 to cancel): 0x");
 fgets(input, DEF_BUF_SIZE / 4, stdin);

 if(input[0] == '-' && atoi(&input[1]) > 0) {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */

 while(strlen(input) &&
 isspace(input[strlen(input) - 1]))
 input[strlen(input) - 1] = '\0';
 if(!strlen(input)) continue;

 for(i = 0; i < strlen(input); i++) {
 if(input[i] >= 'A' && input[i] <= 'F')
 input[i] = input[i] - 'A' + 'a';
 else if((input[i] < '0' || input[i] > '9') &&
 (input[i] < 'a' || input[i] > 'f')) ||
 i >= sizeof(uchar) * 2) {
 printf("Input must be from 0 to 0xFF\n");
 memset(input, '\0', DEF_BUF_SIZE / 4);
 brk = TRUE;
 break;
 } /* else if */
 } /* for */
 if(brk) continue;
 } /* while */

 ado.partition_number = char_to_hex(input);

 memset(input, '\0', DEF_BUF_SIZE / 4);
 while((input[0] < '0' || input[0] > '9')
 && (input[0] < 'a' || input[0] > 'f')) {
 brk = FALSE;
 printf("Enter block ID in hex (< 0 to cancel): 0x");
 fgets(input, DEF_BUF_SIZE / 4, stdin);

 if(input[0] == '-' && atoi(&input[1]) > 0) {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */

 while(strlen(input) &&
 isspace(input[strlen(input) - 1]))
 input[strlen(input) - 1] = '\0';
 if(!strlen(input)) continue;

 for(i = 0; i < strlen(input); i++) {
 if(input[i] >= 'A' && input[i] <= 'F')
 input[i] = input[i] - 'A' + 'a';
 else if((input[i] < '0' || input[i] > '9') &&
 (input[i] < 'a' || input[i] > 'f')) ||
 i >= sizeof(unsigned long long) * 2) {
 printf("Input out of range\n");
 memset(input, '\0', DEF_BUF_SIZE / 4);
 brk = TRUE;
 break;
 } /* else if */
 }
 }
 }
}
```

```

 } /* for */
 if(brk) continue;

 } /* while */

 ado.logical_block_id = char_to_hex(input);
 break;
case 1:
 break;
default:
 assert(!"Unreachable.");
} /* switch */

printf("Issuing IOCTL...\n");
rc = ioctl(fd, STIOC_ALLOW_DATA_OVERWRITE, &ado);

if(rc) {
 printf("Allow data overwrite failed: %d\n", rc);
 goto EXIT_LABEL;
} /* if */

EXIT_LABEL:

 if(input) free(input);
 return rc;
} /* stioc_allow_overwrite() */

```

---

## STIOC\_READ\_POSITION\_EX

This IOCTL returns tape position with support for the short, long, and extended formats. The definitions and data structures that are used for this IOCTL follow. See the READ\_POSITION section of your tape drive's SCSI documentation for details on the short\_data\_format, long\_data\_format, and extended\_data\_format structures.

```

#define RP_SHORT_FORM (0x00)
#define RP_LONG_FORM (0x06)
#define RP_EXTENDED_FORM (0x08)

struct short_data_format {
#ifdef __LITTLE_ENDIAN
 unchar bpew : 1;
 unchar perr : 1;
 unchar lolu : 1;
 unchar rsvd : 1;
 unchar bycu : 1;
 unchar locu : 1;
 unchar eop : 1;
 unchar bop : 1;
#elif defined __BIG_ENDIAN
 unchar bop : 1;
 unchar eop : 1;
 unchar locu : 1;
 unchar bycu : 1; unchar rsvd : 1;
 unchar lolu : 1;
 unchar perr : 1;
 unchar bpew : 1;
#else
 error
#endif
 unchar active_partition;
 char reserved[2];
 unchar first_logical_obj_position[4];
 unchar last_logical_obj_position[4];
 unchar num_buffer_logical_obj[4];
 unchar num_buffer_bytes[4];
 char reserved1;
};

struct long_data_format {
#ifdef __LITTLE_ENDIAN
 unchar bpew : 1;
 unchar rsvd2 : 1;
 unchar lonu : 1;
 unchar mpu : 1;
 unchar rsvd1 : 2;
 unchar eop : 1;
 unchar bop : 1;
#elif defined __BIG_ENDIAN
 unchar bop : 1;
 unchar eop : 1;
 unchar rsvd1 : 2;
 unchar mpu : 1;
 unchar lonu : 1;
 unchar rsvd2 : 1;
 unchar bpew : 1;
#else
 error
#endif
 char reserved[6];
 unchar active_partition;
 unchar logical_obj_number[8];
 unchar logical_file_id[8];
 unchar obsolete[8];
};

```

```

};

struct extended_data_format {
#ifdef __LITTLE_ENDIAN
 unchar bpew : 1;
 unchar perr : 1;
 unchar lolu : 1;
 unchar rsvd : 1;
 unchar bycu : 1;
 unchar locu : 1;
 unchar eop : 1;
 unchar bop : 1;
#elif defined __BIG_ENDIAN
 unchar bop : 1;
 unchar eop : 1;
 unchar locu : 1;
 unchar bycu : 1;
 unchar rsvd : 1;
 unchar lolu : 1;
 unchar perr : 1;
 unchar bpew : 1;
#else
 error
#endif
 unchar active_partition;
 unchar additional_length[2];
 unchar num_buffer_logical_obj[4];
 unchar first_logical_obj_position[8];
 unchar last_logical_obj_position[8];
 unchar num_buffer_bytes[8];
 unchar reserved;
};

struct read_tape_position {
 unchar data_format;
 union {
 struct short_data_format rp_short;
 struct long_data_format rp_long;
 struct extended_data_format rp_extended;
 } rp_data;
};

```

`data_format` is the format in which you want to receive your data, as defined here. It can take the value `RP_SHORT_FORM`, `RP_LONG_FORM`, or `RP_EXTENDED_FORM`. When the IOCTL finishes, data is returned to the corresponding structure within the `rp_data` union. An example of the use of the **STIOC\_READ\_POSITION\_EX** IOCTL is

```

int stioc_read_position_ex(void)
{
 int rc = 0;
 char* input = NULL;
 struct read_tape_position rp = {0};

 printf("Note: only supported on LTO 5 and higher drives\n");
 input = malloc(DEF_BUF_SIZE / 16);
 if(!input) {
 rc = ENOMEM;
 goto EXIT_LABEL;
 } /* if */
 memset(input, '\0', DEF_BUF_SIZE / 16);

 while(input[0] == '\0' || atoi(input) < 0 || atoi(input) > 3) {
 printf("0) Cancel\n");
 printf("1) Short Form\n");
 printf("2) Long Form\n");
 printf("3) Extended Form\n");

 printf("\nPlease select: ");

 fgets(input, DEF_BUF_SIZE / 16, stdin);
 if(!atoi(input)) {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */
 } /* while */

 memset(&rp, '\0', sizeof(struct read_tape_position));

 switch(atoi(input)) {
 case 1:
 rp.data_format = RP_SHORT_FORM;
 break;
 case 2:
 rp.data_format = RP_LONG_FORM;
 break;
 case 3:
 rp.data_format = RP_EXTENDED_FORM;
 break;
 default:
 rc = EINVAL;
 goto EXIT_LABEL;
 } /* switch */

 rc = ioctl(fd, STIOC_READ_POSITION_EX, &rp);

 if(rc) {

```

```

 printf("Cannot get position: %d\n", rc = errno);
 goto EXIT_LABEL;
 } /* if */

 print_read_position_ex(&rp);

EXIT_LABEL:
 if(input) free(input);
 return rc;
} /* stioc_read_position_ex() */

```

---

## STIOC\_LOCATE\_16

This IOCTL sets the tape position by using the long tape format. The definitions and structure that are used for this IOCTL are

```

#define LOGICAL_ID_BLOCK_TYPE (0x00)
#define LOGICAL_ID_FILE_TYPE (0x01)

struct set_tape_position {
 uchar logical_id_type;
 unsigned long long logical_id;
 char reserved[32];
};

```

`logical_id_type` can take the values `LOGICAL_ID_BLOCK_TYPE` or `LOGICAL_ID_FILE_TYPE`. The values specify whether the tape head is located to the block with the specified `logical_id` or to the file with the specified `logical_id`. An example on how to use the **STIOC\_LOCATE\_16** IOCTL follows. The snippet assumes the declaration of global variables `filetype` and `blockid`.

```

int stioc_locate_16(void)
{
 int rc = 0;
 struct set_tape_position pos;

 memset(&pos, '\0', sizeof(struct set_tape_position));
 printf("\nLocating to %s ID %u (0x%08X)...\n",
 filetype ? "File" : "Block", blockid, blockid);

 pos.logical_id_type = filetype;
 pos.logical_id = (long long) blockid;

 rc = ioctl(fd, STIOC_LOCATE_16, &pos);
 return rc;
} /* stioc_locate_16() */

```

---

## STIOC\_QUERY\_BLK\_PROTECTION

This IOCTL queries capability and status of the drive's Logical Block Protection. The structures and defines are

```

#define LBP_DISABLE (0x00)
#define REED_SOLOMON_CRC (0x01)

struct logical_block_protection {
 uchar lbp_capable;
 uchar lbp_method;
 uchar lbp_info_length;
 uchar lbp_w;
 uchar lbp_r;
 uchar rbdp;
 uchar reserved[26];
};

```

The `lbp_capable` is set to True if the drive supports logical block protection, or False otherwise.

A `lbp_method` method of `LBP_DISABLE` indicates that the logical block protection feature is turned off. A value of `REED_SOLOMON_CRC` indicates that logical block protection is used, with a Reed-Solomon cyclical redundancy check algorithm to run the block protection.

The `lbp_w` indicates that logical block protection is run for write commands. The `lbp_r` indicates that logical block protection is run for read commands. The `rbdp` indicates that logical block protection is run for recover buffer data. To use this IOCTL, issue the following call.

```
rc = ioctl(fd, STIOC_QUERY_BLK_PROTECTION, &lbp);
```

---

## STIOC\_SET\_BLK\_PROTECTION

This IOCTL sets status of the drive's Logical Block Protection. All fields are configurable except `lbp_capable` and `reserved`. The structures and defines are the same as for **STIOC\_QUERY\_BLK\_PROTECTION**. To use this IOCTL, issue the following call.

```
rc = ioctl(fd, STIOC_SET_BLK_PROTECTION, &lbp);
```

---

## STIOC\_VERIFY\_TAPE\_DATA

This IOCTL instructs the tape drive to scan the data on its current tape to check for errors. The structure is defined as follows.

```
struct verify_data {
 #if defined __LITTLE_ENDIAN
 unchar fixed : 1;
 unchar bytcmp : 1;
 unchar immed : 1;
 unchar vbf : 1;
 unchar vlbpm : 1;
 unchar vte : 1;
 unchar reserved1 : 2;
 #elif defined __BIG_ENDIAN
 unchar reserved1 : 2;
 unchar vte : 1;
 unchar vlbpm : 1;
 unchar vbf : 1;
 unchar immed : 1;
 unchar bytcmp : 1;
 unchar fixed : 1;
 #else
 error
 #endif
 unchar verify_length[3];
 unchar reserved2[15];
};
```

**vte** instructs the drive to verify from the current tape head position to end of data.

**vlbpm** instructs the drive to verify that the logical block protection method that is specified in the Control Data Protection mode page is used for each block.

If **vbf** is set, then the **verify\_length** field contains the number of filemarks to be traversed, rather than the number of blocks or bytes.

**immed** specifies that status is to be returned immediately after the command descriptor block is validated. Otherwise, the command does not return status until the entire operation finishes.

**bytcmp** is set to 0.

**fixed** indicates a fixed-block length, and that **verify\_length** is interpreted as blocks rather than bytes.

**verify\_length** specifies the length to verify in files, blocks or bytes, depending on the values of the **vbf** and **fixed** fields. If **vte** is set to 1, **verify\_length** is ignored.

An example of the use of **STIOC\_VERIFY\_TAPE\_DATA** is as follows.

```
int stioc_verify()
{
 int rc = 0, i = 0, cont = TRUE, len = 0;
 char* input = NULL;
 struct verify_data* vfy = NULL;

 struct {
 char* desc;
 int idx;
 } table[] = {
 {"Verify to EOD", VFY_VTE},
 {"Verify Logical Block Protection", VFY_VLBPM},
 {"Verify by Filemarks", VFY_VBF},
 {"Return immediately", VFY_IMMED},
 {"Fixed", VFY_FIXED},
 {NULL, 0}
 };

 input = malloc(DEF_BUF_SIZE / 16);
 if(!input) {
 rc = ENOMEM;
 goto EXIT_LABEL;
 } /* if */
 memset(input, '\0', DEF_BUF_SIZE / 16);

 vfy = malloc(sizeof(struct verify_data));
 if(!vfy) {
 rc = ENOMEM;
 goto EXIT_LABEL;
 } /* if */
 memset(vfy, '\0', sizeof(struct verify_data));

 printf("\n");
 for(i = 0; table[i].desc; i++) {
 while(tolower(input[0]) != 'y' && tolower(input[0]) != 'n') {
 printf("%s (y/n/c to cancel)? ", table[i].desc);
 fgets(input, DEF_BUF_SIZE / 16, stdin);
 if(tolower(input[0]) == 'c') {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */
 } /* while */
 } /* for */

 if(tolower(input[0]) == 'y') {
 switch(table[i].idx) {
 case VFY_VTE: vfy->vte = 1; break;
 case VFY_VLBPM: vfy->vlbpm = 1; break;
 case VFY_VBF: vfy->vbf = 1; break;
 case VFY_IMMED: vfy->immed = 1; break;
 default: break;
 } /* switch */
 } /* if */
}
```

```

 memset(input, '\0', DEF_BUF_SIZE / 16);
 } /* for */

 if(!vfy->vte) {
 while(cont) {
 cont = FALSE;

 printf("Verify length in decimal (c to cancel): ");
 fgets(input, DEF_BUF_SIZE / 16, stdin);

 while(strlen(input) && isspace(input[strlen(input)-1]))
 input[strlen(input) - 1] = '\0';

 if(!strlen(input)) {
 cont = TRUE;
 continue;
 } /* if */

 if(tolower(input[0]) == 'c') {
 rc = 0;
 goto EXIT_LABEL;
 } /* if */

 for(i = 0; i < strlen(input); i++) {
 if(!isdigit(input[i])) {
 memset(input, '\0', DEF_BUF_SIZE / 16);
 cont = TRUE;
 } /* if */
 } /* for */

 } /* while */

 len = atoi(input);
 vfy->verify_length[0] = (len >> 16) & 0xFF;
 vfy->verify_length[1] = (len >> 8) & 0xFF;
 vfy->verify_length[2] = len & 0xFF;
 } /* if */

 rc = ioctl(fd, STIOC_VERIFY_TAPE_DATA, &vfy);
 printf("VERIFY TAPE DATA returned %d\n", rc);
 if(rc) printf("errno: %d\n", errno);

EXIT_LABEL:

 if(input) free(input);
 if(vfy) free(vfy);
 return rc;
} /* stioc_verify() */

```

---

## STIOC\_QUERY\_RAO

The IOCTL is used to query the maximum number and size of User Data Segments (UDS) that are supported from tape drive and driver for the wanted uds\_type.

The application calls this IOCTL before the **STIOC\_GENERATE\_RAO** and **STIOC\_RECEIVE\_RAO** IOCTLs are issued. The application uses the return data to limit the number of UDS requested in the **GENERATE\_RAO** IOCTL.

The structure that is defined for this IOCTL is

```

struct query_rao_info{
 char uds_type; /* [IN] 0: UDS WITHOUT GEOMETRY */
 /* 1: UDS WITH GEOMETRY */
 char reserved[7];
 ushort max_uds_number; /* [OUT] Max UDS number supported from drive */
 ushort max_uds_size; /* [OUT] Max single UDS size supported from */
 /* drive in byte */
 ushort max_host_uds_number; /* [OUT] Max UDS number supported from driver */
};

```

An example of the **QUERY\_RAO\_INFO** command is

```

#include <sys/IBMtape.h>
int rc;
struct query_rao_info stQueryRao;

bzero((void *) &stQueryRao, sizeof(struct query_rao_info));

stQueryRao.uds_type = uds_type;

rc = ioctl(fd, STIOC_QUERY_RAO, &stQueryRao);

if(rc)
 printf("STIOC_QUERY_RAO fails with rc: %d\n", rc);
else{
 max_host_uds_num = stQueryRao.max_host_uds_number;
 max_uds_size = stQueryRao.max_uds_size;
}
return rc;

```

---

## STIOC\_GENERATE\_RAO

The IOCTL is called to send a GRAO list to request that the drive generate a Recommended Access Order list. The process method is either 1 or 2 to create a RAO list, and the type of UDS is either with or without the geometry. The `uds_number` must be not larger than `max_host_uds_number` in the **STIOC\_QUERY\_RAO** IOCTL. The application allocates a block of memory with `grao_list_leng (uds_number*sizeof(struct grao_uds_desc)+8)` for the pointer of `grao_list`.

The structure for the **STIOC\_GENERATE\_RAO** IOCTL is

```
struct generate_rao {
 char process; /* [IN] Requested process to generate RAO list */
 /* 0: no reorder UDS and no calculate */
 /* locate time(not currently supported */
 /* by the drive) */
 /* 1: no reorder UDS but calculate locate */
 /* time */
 /* 2: reorder UDS and calculate locate time*/
 char uds_type; /* [IN] 0: UDS_WITHOUT_GEOMETRY */
 /* 1: UDS_WITH_GEOMETRY */
 char reserved1[2];
 uint grao_list_leng; /* [IN] The data length is allocated for GRAO */
/* list */
 char *grao_list; /* [IN] the pointer is allocated to the size */
 /* of grao_list_leng (uds_number */
 /* * sizeof(struct grao_uds_desc) */
 /* + sizeof(struct grao_list_header)) */
 /* and contains the data of GRAO */
 /* parameter list. The uds number is */
 /* less than max_host_uds_number in */
 /* QUERY_RAO ioctl. */
 char reserved2[8];
};
```

The `grao_list` header and UDS segments make up the parameter data and are to be put in the following order.

```
-- List Header
-- UDS Segment Descriptor (first)
.....
-- UDS Segment Descriptor (last)
```

The device driver does not supply the header or UDS segment Descriptor structures. That structure is to be supplied by the application.

Examples of the data structures are

```
struct grao_list_header{
 unchar reserved[4];
 char addl_data[4]; /* additional data */
};

struct grao_uds_desc{
 unchar desc_leng[2]; /* descriptor length */
 char reserved[3];
 char uds_name[10]; /* uds name given by application */
 unchar partition; /* Partition number 0-n to overwrite */
 unchar beginning_loi[8]; /* Beginning logical object ID */
 unchar ending_loi[8]; /* Ending logical object ID */
};
```

A sample of **STIOC\_GENERATE\_RAO** is

```
#include<sys/IBM_tape.h>

int rc;
struct generate_rao grao;
bzero(&grao,sizeof(struct generate_rao));
grao.process=2;
grao.uds_type=uds_type;
grao.grao_list_leng=max_host_uds_num * sizeof(struct grao_uds_desc)
+ sizeof(struct grao_list_header);
if(!(grao.grao_list=malloc(grao.grao_list_leng)))
{
 perror("Failure allocating memory");
 return (errno);
}

memset(grao.grao_list, 0, grao.grao_list_leng);

rc=ioctl(fd,GENERATE_RAO,&grao);

if (rc)
 printf("GENERATE_RAO fails with rc%d\n",rc);
else
 printf("GENERATE_RAO succeeds\n");

free(grao.grao_list);

return rc;
```

## STIOC\_RECEIVE\_RAO

After a **STIOC\_GENERATE\_RAO** IOCTL is completed, the application calls the **STIOC\_RECEIVE\_RAO** IOCTL to receive a recommended access order of UDS from the drive. To avoid a system crash, it is important that the application allocates a large enough block of memory for the `*rrao_list` pointer and notifies the driver of the allocated

size. It is done by indicating the size of the buffer in bytes to the *rrao\_list\_leng* variable as an input to the *receive\_rao\_list* structure.

The structure for the **STIOC\_RECEIVE\_RAO** IOCTL is

```
struct receive_rao_list {
 uint rrao_list_offset; /* [IN] The offset of receive RAO list to */
 /* begin returning data */
 uint rrao_list_leng; /* [IN/OUT] number byte of data length */
 /* [IN] The data length is allocated for RRAO */
 /* list by application the length is */
 /* (max_uds_size * uds number + */
 /* sizeof(struct rrao_list_header) */
 /* max_uds_size is reported in */
 /* sizeof(struct rrao_list_header) */
 /* uds number is the total UDS number */
 /* requested from application in */
 /* GENERATE_RAO ioctl */
 /* [OUT] the data length is actual returned */
 /* in RRAO list from the driver */
 char *rrao_list; /* [IN/OUT] the data pointer of RRAO list */
 char reserved[8];
};
```

The sample code is

```
#include <sys/IBMtape.h>

int rc;
struct receive_rao_list rrao;

bzero(&rrao, sizeof(struct receive_rao_list));
rrao.rrao_list_offset=0;
rrao.rrao_list_leng= max_host_uds_num * max_uds_size + 8;
/* 8 is the header of rrao list */

if (!(rrao.rrao_list=malloc(rrao.rrao_list_leng)))
{
 perror("Failure allocating memory");
 return (errno);
}

memset(rrao.rrao_list, 0, rrao.rrao_list_leng);

rc=ioctl(fd, STIOC_RECEIVE_RAO, &rrao);

if (rc)
 printf("STIOC_RECEIVE_RAO fails with rc %d\n", rc);
else
 printf("rrao_list_leng %d\n", rrao.rrao_list_leng);

free(rrao.rrao_list);

return rc;
```

---

## STIOC\_SET\_SPDEV

With the latest *lin\_tape* versions, the IBMSpecial device is created. It allows the use of *ioctl*s for preemption purposes. Applications must use it cautiously and manage persistent reservation properly.

This *ioctl* is for usage through IBMSpecial open handle only. It sets the drive that processes the command requests, and to do so it needs the serial number of the drive as input. If */dev/IBMSpecial* is not created, it is not supported.

The data structure is

```
#define DD_MAX_DEVICE_SERIAL 36
struct sp_dev{
 char device_serial[DD_MAX_DEVICE_SERIAL];
};
```

An example of the **STIOC\_SET\_SPDEV** command is

```
#include <sys/IBM_tape.h>
struct sp_dev spd;

setDriveSN(spd.device_serial);
if (!ioctl (fd, STIOC_SET_SPDEV, &spd)) {
 printf ("The STIOC_SET_SPDEV ioctl succeeded\n");
}
else {
 perror ("The STIOC_SET_SPDEV ioctl failed the drive to work with was not set");
}
```

When the **STIOC\_SET\_SPDEV** *ioctl* succeeds, it is possible to send any of these *ioctl*s to the drive previously set and identified by serial number:

**STIOC\_READ\_RESERVEKEYS**, **STIOC\_READ\_RESERVATIONS**, **STIOC\_REGISTER\_KEY**, **STIOC\_REMOVE\_REGISTRATION**, **STIOC\_CLEAR\_ALL\_REGSITRATION**.

---

## Tape drive compatibility IOCTL operations



The following IOCTL commands help provide compatibility for previously compiled programs. Where practical, such programs must be recompiled to use the preferred IOCTL commands in the IBMtape device driver.

- [MTIOCTOP](#)
- [MTIOCGET](#)
- [MTIOCPOS](#)

---

## MTIOCTOP

This IOCTL command is similar in function to the st **MTIOCTOP** command. It is provided as a convenience for precompiled programs that call that IOCTL command. Refer to `/usr/include/sys/mtio.h` or `/usr/include/linux/mtio.h` for information on the **MTIOCTOP** command.

---

## MTIOCGET

This IOCTL command is similar in function to the st **MTIOCGET** command. It is provided as a convenience for precompiled programs that call that IOCTL command. Refer to `/usr/include/sys/mtio.h` or `/usr/include/linux/mtio.h` for information on the **MTIOCGET** command.

---

## MTIOCPOS

This IOCTL command is similar in function to the st **MTIOCPOS** command. It is provided as a convenience for precompiled programs that call that IOCTL command. Refer to `/usr/include/sys/mtio.h` or `/usr/include/linux/mtio.h` for information on the **MTIOCPOS** command.

---

## Medium changer IOCTL operations

This chapter describes the IOCTL commands that provide access and control of the SCSI medium changer functions. These IOCTL operations can be issued to the medium changer special file, such as `IBMchanger0`.

The following IOCTL commands are supported.

**SMCIOCT\_ELEMENT\_INFO**  
Obtain the device element information.

**SMCIOCT\_MOVE\_MEDIUM**  
Move a cartridge from one element to another element.

**SMCIOCT\_EXCHANGE\_MEDIUM**  
Exchange a cartridge in an element with another cartridge.

**SMCIOCT\_POS\_TO\_ELEM**  
Move the robot to an element.

**SMCIOCT\_INIT\_ELEM\_STAT**  
Issue the **SCSI Initialize Element Status** command.

**SMCIOCT\_INIT\_ELEM\_STAT\_RANGE**  
Issue the **SCSI Initialize Element Status with Range** command.

**SMCIOCT\_INVENTORY**  
Return the information about the four element types.

**SMCIOCT\_LOAD\_MEDIUM**  
Load a cartridge from a slot into the drive.

**SMCIOCT\_UNLOAD\_MEDIUM**  
Unload a cartridge from the drive and return it to a slot.

**SMCIOCT\_PREVENT\_MEDIUM\_REMOVAL**  
Prevent medium removal by the operator.

**SMCIOCT\_ALLOW\_MEDIUM\_REMOVAL**  
Allow medium removal by the operator.

**SMCIOCT\_READ\_ELEMENT\_DEVIDS**  
Return the device id element descriptors for drive elements.

- [SCSI IOCTL commands](#)

---

## SCSI IOCTL commands

These IOCTL commands and their associated structures are defined in the `IBM_tape.h` header file, which can be found in `/usr/include/sys` after IBMtape is installed. The `IBM_tape.h` header file is included in the corresponding C program by using the functions.

- [SMCIOCT\\_ELEMENT\\_INFO](#)
- [SMCIOCT\\_MOVE\\_MEDIUM](#)
- [SMCIOCT\\_EXCHANGE\\_MEDIUM](#)
- [SMCIOCT\\_POS\\_TO\\_ELEM](#)
- [SMCIOCT\\_INIT\\_ELEM\\_STAT](#)
- [SMCIOCT\\_INIT\\_ELEM\\_STAT\\_RANGE](#)

- [SMCIOC\\_INVENTORY](#)
- [SMCIOC\\_LOAD\\_MEDIUM](#)
- [SMCIOC\\_UNLOAD\\_MEDIUM](#)
- [SMCIOC\\_PREVENT\\_MEDIUM\\_REMOVAL](#)
- [SMCIOC\\_ALLOW\\_MEDIUM\\_REMOVAL](#)
- [SMCIOC\\_READ\\_ELEMENT\\_DEVIDS](#)

---

## SMCIOC\_ELEMENT\_INFO

This IOCTL command obtains the device element information.

The data structure is

```
struct element_info {
 ushort robot_addr; /* first robot address */
 ushort robots; /* number of medium transport elements */
 ushort slot_addr; /* first medium storage element address */
 ushort slots; /* number of medium storage elements */
 ushort ie_addr; /* first import/export element address */
 ushort ie_stations; /* number of import/export elements */
 ushort drive_addr; /* first data-transfer element address */
 ushort drives; /* number of data-transfer elements */
};
```

An example of the **SMCIOC\_ELEMENT\_INFO** command is

```
#include <sys/IBM_tape.h>
struct element_info element_info;
if (!ioctl (smcfd, SMCIOC_ELEMENT_INFO, &element_info)) {
 printf ("The SMCIOC_ELEMENT_INFO ioctl succeeded\n");
 printf ("\nThe element information data is:\n");
 dump_bytes ((uchar *) &element_info, sizeof (struct element_info));
}
else {
 perror ("The SMCIOC_ELEMENT_INFO ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_MOVE\_MEDIUM

This IOCTL command moves a cartridge from one element to another element.

The data structure is

```
struct move_medium {
 ushort robot; /* robot address */
 ushort source; /* move from location */
 ushort destination; /* move to location */
 char invert; /* invert before placement bit */
};
```

An example of the **SMCIOC\_MOVE\_MEDIUM** command is

```
#include <sys/IBM_tape.h>
struct move_medium move_medium;
move_medium.robot = 0;
move_medium.invert = 0;
move_medium.source = source;
move_medium.destination = dest;
if (!ioctl (smcfd, SMCIOC_MOVE_MEDIUM, &move_medium))
 printf ("The SMCIOC_MOVE_MEDIUM ioctl succeeded\n");
else {
 perror ("The SMCIOC_MOVE_MEDIUM ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_EXCHANGE\_MEDIUM

This IOCTL command exchanges a cartridge in an element with another cartridge. This command is equivalent to two **SCSI Move Medium** commands. The first moves the cartridge from the source element to the destination1 element. The second moves the cartridge that was previously in the destination1 element to the destination 2 element. This function is available only in the IBM® 3584 UltraScalable tape library. The destination2 element can be the same as the source element.

The input data structure is

```
struct exchange_medium {
 ushort robot; /* robot address */
 ushort source; /* source address for exchange */
 ushort destination1; /* first destination address for exchange */
 ushort destination2; /* second destination address for exchange */
 char invert1; /* invert before placement into destination1 */
 char invert2; /* invert before placement into destination2 */
};
```

An example of the **SMCIOC\_EXCHANGE\_MEDIUM** command is

```
#include <sys/IBM_tape.h>
struct exchange_medium exchange_medium;
exchange_medium.robot = 0;
exchange_medium.invert1 = 0;
exchange_medium.invert2 = 0;
exchange_medium.source = 32; /* slot 32 */
exchange_medium.destination1 = 16; /* drive address 16 */
exchange_medium.destination2 = 35; /* slot 35 */

/* exchange cartridge in drive address 16 with cartridge from */
/* slot 32 and return the cartridge currently in the drive to */
/* slot 35 */
if (!ioctl (smcfd, SMCIOC_EXCHANGE_MEDIUM, &exchange_medium))
 printf("The SMCIOC_EXCHANGE_MEDIUM ioctl succeeded\n");
else {
 perror ("The SMCIOC_EXCHANGE_MEDIUM ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_POS\_TO\_ELEM

This IOCTL command moves the robot to an element.

The input data structure is

```
struct pos_to_elem {
 ushort robot; /* robot address */
 ushort destination; /* move to location */
 char invert; /* invert before placement bit */
};
```

An example of the **SMCIOC\_POS\_TO\_ELEM** command is

```
#include <sys/IBM_tape.h>
struct pos_to_elem pos_to_elem;
pos_to_elem.robot = 0;
pos_to_elem.invert = 0;
pos_to_elem.destination = dest;
if (!ioctl (smcfd, SMCIOC_POS_TO_ELEM, &pos_to_elem))
 printf ("The SMCIOC_POS_TO_ELEM ioctl succeeded\n");
else {
 perror ("The SMCIOC_POS_TO_ELEM ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_INIT\_ELEM\_STAT

This IOCTL command instructs the medium changer robotic device to issue the **SCSI Initialize Element Status** command.

There is no associated data structure.

An example of the **SMCIOC\_INIT\_ELEM\_STAT** command is

```
#include <sys/IBM_tape.h>
if (!ioctl (smcfd, SMCIOC_INIT_ELEM_STAT, NULL))
 printf ("The SMCIOC_INIT_ELEM_STAT ioctl succeeded\n");
else {
 perror ("The SMCIOC_INIT_ELEM_STAT ioctl failed");
 smcioc_request_sense();
}
```

---

## SMCIOC\_INIT\_ELEM\_STAT\_RANGE

This IOCTL command issues the **SCSI Initialize Element Status with Range** command and audits specific elements in a library by specifying the starting element address and number of elements. Use the **SMCIOC\_INIT\_ELEM\_STAT** IOCTL to audit all elements.

The data structure is

```
struct element_range {
 ushort element_address; /* starting element address */
 ushort number_elements; /* number of elements */
};
```

An example of the **SMCIOC\_INIT\_ELEM\_STAT\_RANGE** command is

```
#include <sys/IBM_tape.h>
struct element_range elements;
/* audit slots 32 to 36 */
elements.element_address = 32;
elements.number_elements = 5;
if (!ioctl (smcfd, SMCIOC_INIT_ELEM_STAT_RANGE, &elements))
```

```

 printf ("The SMCIIOC_INIT_ELEM_STAT_RANGE ioctl succeeded\n");
 else {
 perror ("The SMCIIOC_INIT_ELEM_STAT_RANGE ioctl failed");
 smcioc_request_sense();
 }
}

```

Note: Use the **SMCIIOC\_INVENTORY** IOCTL command to obtain the current version after this IOCTL command is issued.

## SMCIIOC\_INVENTORY

This IOCTL command returns the information about the four element types. The software application processes the input data (the number of elements about which it requires information). Then, it allocates a buffer large enough to hold the output for each element type.

The input data structure is

```

struct element_status {
 ushort address; /* element address */
 uint :2, /* reserved */
 inenab :1, /* media into changer's scope */
 exenab :1, /* media out of changer's scope */
 access :1, /* robot access allowed */
 except :1, /* abnormal element state */
 impexp :1, /* import/export placed by operator or robot */
 full :1, /* element contains medium */
 unchar resvd1; /* reserved */
 unchar asc; /* additional sense code */
 unchar ascq; /* additional sense code qualifier */
 uint notbus :1, /* element not on same bus as robot */
 :1, /* reserved */
 idvalid :1, /* element address valid */
 luvalid :1, /* logical unit valid */
 :1, /* reserved */
 lun :3; /* logical unit number */
 unchar scsi; /* SCSI bus address */
 unchar resvd2; /* reserved */
 uint svalid :1, /* element address valid */
 invert :1, /* medium inverted */
 :6; /* reserved */
 ushort source; /* source storage element address */
 unchar volume[36]; /* primary volume tag */
 unchar resvd3[4]; /* reserved */
};

struct inventory {
 struct element_status *robot_status; /* medium transport elem pgs */
 struct element_status *slot_status; /* medium storage elem pgs */
 struct element_status *ie_status; /* import/export elem pgs */
 struct element_status *drive_status; /* data-transfer elem pgs */
};

```

An example of the **SMCIIOC\_INVENTORY** command is

```

#include <sys/IBM_tape.h>
ushort i;
struct element_info element_info;
struct element_status robot_status[1];
struct element_status slot_status[20];
struct element_status ie_status[1];
struct element_status drive_status[1];
struct inventory inventory;
bzero((caddr_t)robot_status,sizeof(struct element_status));
for (i=0;i<20;i++)
 bzero((caddr_t)(&slot_status[i]),sizeof(struct element_status));
bzero((caddr_t)ie_status,sizeof(struct element_status));
bzero((caddr_t)drive_status,sizeof(struct element_status));
smcioc_element_info(&element_info);
inventory.robot_status = robot_status;
inventory.slot_status = slot_status;
inventory.ie_status = ie_status;
inventory.drive_status = drive_status;
if (!ioctl (smcfd, SMCIIOC_INVENTORY, &inventory)) {
 printf ("\nThe SMCIIOC_INVENTORY ioctl succeeded\n");
 printf ("\nThe robot status pages are:\n");
 for (i = 0; i<element_info.robots; i++) {
 dump_bytes ((unchar *) (robot_status[i]), sizeof (struct
 element_status));
 printf ("\n--- more ---");
 getchar();
 }
 printf ("\nThe slot status pages are:\n");
 for (i = 0; i<element_info.slots; i++) {
 dump_bytes ((unchar *) (slot_status[i]), sizeof (struct
 element_status));
 printf ("\n--- more ---");
 getchar();
 }
 printf ("\nThe ie status pages are:\n");
 for (i = 0; i<element_info.ie_stations; i++) {
 dump_bytes ((unchar *) (ie_status[i]), sizeof (struct
 element_status));
 printf ("\n--- more ---");
 getchar();
 }
}

```

```

 }
 printf ("\nThe drive status pages are:\n");
 for (i = 0; i<element_info.drives; i++) {
 dump_bytes ((uchar *) (drive_status[i]), sizeof (struct element_status));
 printf ("\n--- more ---");
 getchar();
 }
}
else {
 perror ("The SMCIIOC_INVENTORY ioctl failed");
 smcioc_request_sense();
}
}

```

---

## SMCIIOC\_LOAD\_MEDIUM

This IOCTL command loads a tape from a specific slot into the drive. Or, it loads from the first full slot into the drive if the slot address is specified as zero.

An example of the **SMCIIOC\_LOAD\_MEDIUM** command is

```

#include <sys/IBM_tape.h>
/* load cartridge from slot 3 */
if (ioctl (tapefd, SMCIIOC_LOAD_MEDIUM,3)) {
 printf ("IOCTL failure. errno=%d\n",errno);
 exit(1);
}
/* load first cartridge from magazine */
if (ioctl (tapefd, SMCIIOC_LOAD_MEDIUM,0)) {
 printf ("IOCTL failure. errno=%d\n",errno);
 exit(1);
}

```

---

## SMCIIOC\_UNLOAD\_MEDIUM

This IOCTL command moves a tape from the drive and returns it to a specific slot. Or, it moves a tape to the first empty slot in the magazine if the slot address is specified as zero. An **unload/offline** command must be sent to the tape first, otherwise, this IOCTL command fails with *errno* EIO.

An example of the **SMCIIOC\_UNLOAD\_MEDIUM** command is

```

#include <sys/IBM_tape.h>
/* unload cartridge to slot 3 */
if (ioctl (tapefd, SMCIIOC_UNLOAD_MEDIUM,3)) {
 printf ("IOCTL failure. errno=%d\n",errno);
 exit(1);
}
/* unload cartridge to first empty slot in magazine */
if (ioctl (tapefd, SMCIIOC_UNLOAD_MEDIUM,0)) {
 printf ("IOCTL failure. errno=%d\n",errno);
 exit(1);
}

```

---

## SMCIIOC\_PREVENT\_MEDIUM\_REMOVAL

This IOCTL command prevents an operator from removing medium from the device until the **SMCIIOC\_ALLOW\_MEDIUM\_REMOVAL** command is issued or the device is reset. There is no associated data structure.

An example of the **SMCIIOC\_PREVENT\_MEDIUM\_REMOVAL** command is

```

#include <sys/IBM_tape.h>
if (!ioctl (smcfd, SMCIIOC_PREVENT_MEDIUM_REMOVAL, NULL))
 printf ("The SMCIIOC_PREVENT_MEDIUM_REMOVAL ioctl succeeded\n");
else {
 perror ("The SMCIIOC_PREVENT_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}

```

---

## SMCIIOC\_ALLOW\_MEDIUM\_REMOVAL

This IOCTL command allows an operator to remove medium from the device. This command is normally used after an **SMCIIOC\_PREVENT\_MEDIUM\_REMOVAL** command to restore the device to the default state. There is no associated data structure.

An example of the **SMCIIOC\_ALLOW\_MEDIUM\_REMOVAL** command is

```

#include <sys/IBM_tape.h>
if (!ioctl (smcfd, SMCIIOC_ALLOW_MEDIUM_REMOVAL, NULL))
 printf ("The SMCIIOC_ALLOW_MEDIUM_REMOVAL ioctl succeeded\n");
else {
 perror ("The SMCIIOC_ALLOW_MEDIUM_REMOVAL ioctl failed");
 smcioc_request_sense();
}

```

## SMCIOC\_READ\_ELEMENT\_DEVIDS

This IOCTL command issues the **SCSI Read Element Status** command with the device ID(DVCID) bit set and returns the element descriptors for the data transfer elements. The `element_address` field specifies the starting address of the first data transfer element. The `number_elements` field specifies the number of elements to return. The application must allocate a return buffer large enough for the number of elements that are specified in the input structure.

The input data structure is

```
struct read_element_devids {
 ushort element_address; /* starting element address */
 ushort number_elements; /* number of elements */
 struct element_devid *drive_devid; /* data transfer element pages */
};
```

The output data structure is

```
struct element_devid {
 ushort address; /* element address */
 uint :4, /* reserved */
 access :1, /* robot access allowed */
 except :1, /* abnormal element state */
 :1, /* reserved */
 full :1; /* element contains medium */
 unchar resvd1; /* reserved */
 unchar asc; /* additional sense code */
 unchar ascq; /* additional sense code qualifier */
 uint notbus :1, /* element not on same bus as robot */
 :1, /* reserved */
 idvalid :1, /* element address valid */
 luvalid :1, /* logical unit valid */
 :1, /* reserved */
 lun :3; /* logical unit number */
 unchar scsi; /* scsi bus address */
 unchar resvd2; /* reserved */
 uint svalid :1, /* element address valid */
 invert :1, /* medium inverted */
 :6; /* reserved */
 ushort source; /* source storage element address */
 uint :4, /* reserved */
 code_set :4; /* code set X'2' is all ASCII identifier*/
 uint :4, /* reserved */
 ident_type :4; /* identifier type */
 unchar resvd3; /* reserved */
 unchar ident_len; /* identifier length */
 unchar identifier[36]; /* device identification */
};
```

An example of the **SMCIOC\_READ\_ELEMENT\_DEVIDS** command is

```
#include <sys/IBM_tape.h>
int smcioc_read_element_devids() {
 int i;
 struct element_devid *elem_devid, *elem;
 struct read_element_devids devids;
 struct element_info element_info;
 if (ioctl(fd, SMCIOC_ELEMENT_INFO, &element_info)) return errno;
 if (element_info.drives) {
 elem_devid = malloc(element_info.drives
 * sizeof(struct element_devid));
 if (elem_devid == NULL) {
 errno = ENOMEM;
 return errno;
 }
 bzero((caddr_t)elem_devid, element_info.drives
 * sizeof(struct element_devid));
 devids.drive_devid = elem_devid;
 devids.element_address = element_info.drive_addr;
 devids.number_elements = element_info.drives;
 printf("Reading element device ids?\n");
 if (ioctl(fd, SMCIOC_READ_ELEMENT_DEVIDS, &devids)) {
 free(elem_devid);
 return errno;
 }
 elem = elem_devid;
 for (i = 0; i < element_info.drives; i++, elem++) {
 printf("\nDrive Address %d\n", elem->address);
 if (elem->except)
 printf(" Drive State Abnormal\n");
 else
 printf(" Drive State Normal\n");
 if (elem->asc == 0x81 && elem->ascq == 0x00)
 printf(" ASC/ASCQ %02X%02X (Drive Present)\n",
 elem->asc, elem->ascq);
 else if (elem->asc == 0x82 && elem->ascq == 0x00)
 printf(" ASC/ASCQ %02X%02X (Drive Not Present)\n",
 elem->asc, elem->ascq);
 else
 printf(" ASC/ASCQ %02X%02X\n",
 elem->asc, elem->ascq);
 if (elem->full)
 printf(" Media Present Yes\n");
 else
```

```

 printf(" Media Present No\n");
 if (elem->access)
 printf(" Robot Access Allowed Yes\n");
 else
 printf(" Robot Access Allowed No\n");
 if (elem->svalid)
 printf(" Source Element Address %d\n",
 elem->source);
 else
 printf(" Source Element Address Valid No\n");
 if (elem->invert)
 printf(" Media Inverted Yes\n");
 else
 printf(" Media Inverted No\n");
 if (elem->notbus)
 printf(" Same Bus as Medium Changer No\n");
 else
 printf(" Same Bus as Medium Changer Yes\n");
 if (elem->idvalid)
 printf(" SCSI Bus Address %d\n",elem->scsi);
 else
 printf(" SCSI Bus Address Valid No\n");
 if (elem->luvalid)
 printf(" Logical Unit Number %d\n",elem->lun);
 else
 printf(" Logical Unit Number Valid No\n");
 printf(" Device ID %0.36s\n",
 elem->identifier);
}
else {
 printf("\nNo drives found in element information\n");
}
free(elem_devid);
return errno;
}

```

## Return codes

This chapter describes error codes that are generated by IBMTape when an error occurs during an operation. On error, the operation returns negative one (-1), and the external variable *errno* is set to one of the listed error codes. *Errno* values are defined in */usr/include/errno.h* (and other files that it includes). Application programs must include *errno.h* to interpret the return codes.

Note: For error code EIO, an application can retrieve more information from the device itself. Issue the **STIOCQRYSENSE** IOCTL command when the *sense\_type* equals LASTERROR, or the **SIOC\_REQSENSE** IOCTL command, to retrieve sense data. Then, analyze the sense data by using the appropriate hardware or SCSI reference for that device.

- [General error codes](#)
- [Open error codes](#)
- [Close error codes](#)
- [Read error codes](#)
- [Write error codes](#)
- [IOCTL error codes](#)

## General error codes

The following codes apply to all operations.

|             |                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EBUSY]     | An excessively busy state was encountered in the device.                                                                                                                                                                                                                                                                                                            |
| [EFAULT]    | A memory failure occurred due to an invalid pointer or address.                                                                                                                                                                                                                                                                                                     |
| [EIO]       | An error due to one of the following conditions: <ul style="list-style-type: none"> <li>• An unrecoverable media error was detected by the device.</li> <li>• The device was not ready for operation or a tape was not in the drive.</li> <li>• The device did not respond to SCSI selection.</li> <li>• A bad file descriptor was passed to the device.</li> </ul> |
| [ENOMEM]    | Insufficient memory was available for an internal memory operation.                                                                                                                                                                                                                                                                                                 |
| [ENXIO]     | The device was not configured and is not receiving requests.                                                                                                                                                                                                                                                                                                        |
| [EPERM]     | The process does not have permission to run the desired function.                                                                                                                                                                                                                                                                                                   |
| [ETIMEDOUT] | A command timed out in the device.                                                                                                                                                                                                                                                                                                                                  |

## Open error codes

The following codes apply to open operations.

|          |                                                                                                      |
|----------|------------------------------------------------------------------------------------------------------|
| [EACCES] | The open requires write access when the cartridge loaded in the drive is physically write-protected. |
| [EAGAIN] | The device was already open when an open was attempted.                                              |
| [EBUSY]  | The device was reserved by another initiator or an excessively busy state was encountered.           |
|          |                                                                                                      |

|          |                                                                                                                                                                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EINVAL] | The operation that is requested has invalid parameters or an invalid combination of parameters, or the device is rejecting <b>open</b> commands.                                                                                                                                         |
| [EIO]    | An I/O error occurred that indicates a failure to operate the device. Run failure analysis.                                                                                                                                                                                              |
| [ENOMEM] | Insufficient memory was available for an internal memory operation.                                                                                                                                                                                                                      |
| [EPERM]  | One of the following situations occurred: <ul style="list-style-type: none"> <li>• An open operation with the O_RDWR or O_WRONLY flag was attempted on a write-protected tape.</li> <li>• A write operation was attempted on a device that was opened with the O_RDONLY flag.</li> </ul> |

## Close error codes

The following codes apply to close operations.

|          |                                                                                                                                                                                    |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EBUSY]  | The SCSI subsystem was busy.                                                                                                                                                       |
| [EFAULT] | Memory reallocation failed.                                                                                                                                                        |
| [EIO]    | A command that is issued during close, such as a <b>rewind</b> command, failed because the device was not ready. An I/O error occurred during the operation. Run failure analysis. |

## Read error codes

The following codes apply to read operations.

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EFAULT] | Failure copying from user to kernel space or vice versa.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| [EINVAL] | One of the following situations occurred: <ul style="list-style-type: none"> <li>• The operation that is requested has invalid parameters or an invalid combination of parameters.</li> <li>• The number of bytes requested in the read operation was not a multiple of the block size for a fixed block transfer.</li> <li>• The number of bytes requested in the read operation was greater than the maximum size allowed by the device for variable block transfers.</li> <li>• A read for multiple fixed odd-byte-count blocks was issued.</li> </ul> |
| [ENOMEM] | One of the following situations occurred: <ul style="list-style-type: none"> <li>• The number of bytes requested in the read operation of a variable block record was less than the size of the block. This error is known as an overlength condition.</li> <li>• Insufficient memory was available for an internal memory operation.</li> </ul>                                                                                                                                                                                                          |
| [EPERM]  | A read operation was attempted on a device that was opened with the O_WRONLY flag.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

## Write error codes

The following codes apply to write operations.

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EFAULT] | Failure copying from user to kernel space or vice versa.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| [EINVAL] | One of the following conditions occurred: <ul style="list-style-type: none"> <li>• The operation that is requested has invalid parameters or an invalid combination of parameters.</li> <li>• The number of bytes requested in the write operation was not a multiple of the block size for a fixed block transfer.</li> <li>• The number of bytes requested in the write operation was greater than the maximum block size allowed by the device for variable block transfers.</li> </ul> |
| [EIO]    | The physical end of the medium was detected, or it is a general error that indicates a failure to write to the device. Perform failure analysis.                                                                                                                                                                                                                                                                                                                                           |
| [ENOMEM] | Insufficient memory was available for an internal memory operation.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [ENOSPC] | A write operation failed because it reached the early warning mark. This error code is returned only one time when the early warning is reached and trailer_labels is set to true. A write operation was attempted after the device reached the logical end of the medium and trailer_labels were set to false.                                                                                                                                                                            |
| [EPERM]  | A write operation was attempted on a write protected tape.                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## IOCTL error codes

The following codes apply to IOCTL operations.

|          |                                                                                                                                                                                                                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [EBUSY]  | SCSI subsystem was busy.                                                                                                                                                                                                                                                                                                                 |
| [EFAULT] | Failure copying from user to kernel space or vice versa.                                                                                                                                                                                                                                                                                 |
| [EINVAL] | The operation that is requested has invalid parameters or an invalid combination of parameters. This error code also results if the IOCTL command is not supported by the device. For example, if you are attempting to issue tape drive IOCTL commands to a SCSI medium changer. An invalid or nonexistent IOCTL command was specified. |
| [EIO]    | An I/O error occurred during the operation. Run failure analysis.                                                                                                                                                                                                                                                                        |
| [ENOMEM] | Insufficient memory was available for an internal memory operation.                                                                                                                                                                                                                                                                      |
| [ENOSYS] | The underlying function for this IOCTL command does not exist on this device. (Other devices might support the function.)                                                                                                                                                                                                                |
| [EPERM]  | An operation that modifies the media was attempted on a write-protected tape or a device that was opened with the O_RDONLY flag.                                                                                                                                                                                                         |



---

## Windows tape device drivers

- [Windows programming interface](#)
- [Event log](#)

---

## Windows programming interface

The programming interface conforms to the standard Microsoft Windows Server 2003, Windows Server 2008, and Windows Server 2012 tape device drivers interface. It is detailed in the Microsoft Developer Network (MSDN) Software Development Kit (SDK) and Driver Development Kit (DDK). Common documentation for these similar devices are indicated by 200x.

Windows IBMTape is conformed by two sets of device drivers,

- ibmtpxxx.sys, which supports the IBM® TotalStorage™ or Magstar® tape drives, where
  - ibmtp2k3.sys, ibmtpbs2k3.sys, ibmtpft2k3.sys are used for Windows Server 2003
  - ibmtp2k8.sys, ibmtpbs2k8.sys, ibmtpft2k8.sys are used for Windows Server 2008
  - ibmtp2k12.sys, ibmtpbs2k12.sys, ibmtpft2k12.sys are used for Windows Server 2012
- ibmcgxxx.sys, which supports the IBM TotalStorage or Magstar medium changer, where
  - ibmcg2k3.sys, ibmcgbs2k3.sys, ibmcgft2k3.sys are used for Windows Server 2003
  - ibmcg2k8.sys, ibmcgbs2k8.sys, ibmcgft2k8.sys are used for Windows Server 2008
  - ibmcg2k12.sys, ibmcgbs2k12.sys, ibmcgft2k12.sys are used for Windows Server 2012

The programming interface conforms to the standard Microsoft Windows 200x tape device driver interface. It is detailed in the Microsoft Developer Network (MSDN) Software Development Kit (SDK), and Driver Development Kit (DDK).

- [User-callable entry points](#)
- [Tape Media Changer driver entry points](#)
- [Medium Changer IOCTLs](#)
- [Preempt reservation](#)
- [Vendor-specific \(IBM\) device IOCTLs for DeviceIoControl](#)
- [Variable and fixed block read/write processing](#)

---

## User-callable entry points

The following user-callable tape driver entry points are supported under ibmtpxxx.sys.

- CreateFile
- CloseHandle
- DeviceIoControl
- EraseTape
- GetTapeParameters
- GetTapePosition
- GetTapeStatus
- PrepareTape
- ReadFile
- SetTapeParameters
- SetTapePosition
- WriteFile
- WriteTapemark

---

## Tape Media Changer driver entry points

If the Removable Storage Manager is stopped, then the following user-callable tape media changer driver entry points are supported under ibmcgxxx.sys:

- CreateFile
- CloseHandle
- DeviceIoControl

Users who want to write application programs to issue commands to IBM® TotalStorage™ device drivers must obtain a license to the MSDN and the Microsoft Visual C++ Compiler. Users also need access to IBM hardware reference manuals for IBM TotalStorage devices.

Programs that access the IBM TotalStorage device driver must complete the following steps:

1. Include the following files in the application.

```
#include <ntddscsi.h>
#include <ntddchgr.h>
#include <ntddtape.h> /* Modified as indicated below */
```

2. Add the following lines to ntddtape.h.

```
#define LB_ACCESS FILE_READ_ACCESS | FILE_WRITE_ACCESS
#define M_MTI(x) CTL_CODE(IOCTL_BASE+2,x,METHOD_BUFFERED, LB_ACCESS)
```

```

#define IOCTL_TAPE_OBTAIN_SENSE CTL_CODE(IOCTL_TAPE_BASE, 0x0819,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_OBTAIN_VERSION CTL_CODE(IOCTL_TAPE_BASE, 0x081a,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_LOG_SELECT CTL_CODE(IOCTL_TAPE_BASE, 0x081c,
METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_TAPE_LOG_SENSE CTL_CODE(IOCTL_TAPE_BASE, 0x081d,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_LOG_SENSE10 CTL_CODE(IOCTL_TAPE_BASE, 0x0833,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_ENH_TAPE_LOG_SENSE10 CTL_CODE(IOCTL_TAPE_BASE, 0x0835, METHOD_BUFFERED,
FILE_READ_ACCESS)
#define IOCTL_TAPE_REPORT_MEDIA_DENSITY CTL_CODE(IOCTL_TAPE_BASE, 0x081e,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_OBTAIN_MTDEVICE (M_MTI(16))
#define IOCTL_CREATE_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0826, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_QUERY_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0825, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_SET_ACTIVE_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0827, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_QUERY_DATA_SAFE_MODE CTL_CODE(IOCTL_TAPE_BASE, 0x0823, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_SET_DATA_SAFE_MODE CTL_CODE(IOCTL_TAPE_BASE, 0x0824, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_ALLOW_DATA_OVERWRITE CTL_CODE(IOCTL_TAPE_BASE, 0x0828, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_SET_PEW_SIZE
CTL_CODE(IOCTL_TAPE_BASE, 0x082C, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_QUERY_PEW_SIZE
CTL_CODE(IOCTL_TAPE_BASE, 0x082B, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_VERIFY_TAPE_DATA
CTL_CODE(IOCTL_TAPE_BASE, 0x082A, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_QUERY_RAO_INFO CTL_CODE(IOCTL_TAPE_BASE, 0x082E, METHOD_BUFFERED,
FILE_READ_ACCESS)
#define IOCTL_GENERATE_RAO CTL_CODE(IOCTL_TAPE_BASE, 0x082F, METHOD_BUFFERED,
FILE_READ_ACCESS)
#define IOCTL_RECEIVE_RAO CTL_CODE(IOCTL_TAPE_BASE, 0x0834, METHOD_BUFFERED,
FILE_READ_ACCESS)

```

- [CreateFile](#)
- [ReadFile](#)
- [WriteFile](#)
- [Write Tapemark](#)
- [SetTapePosition](#)
- [GetTapePosition](#)
- [SetTapeParameters](#)
- [GetTapeParameters](#)
- [PrepareTape](#)
- [EraseTape](#)
- [GetTapeStatus](#)
- [DeviceIoControl](#)

---

## CreateFile

The CreateFile entry point is called to make the driver and device ready for input/output (I/O). Installing the driver in non-exclusive mode allows several handles to the same TotalStorage device. If the driver was installed in exclusive mode, by default only one active handle is allowed to a given TotalStorage device. However, if more than one handle is needed for the same device, the dwCreationDisposition parameter can be set to OPEN\_ALWAYS in CreateFile() function to create an extra handle. By using this flag the resulting handle has limited functions. The driver allows the following IOCTLs only:

```

IOCTL_STORAGE_PERSISTENT_RESERVE_IN
IOCTL_STORAGE_PERSISTENT_RESERVE_OUT
IOCTL_SCSI_PASS_THROUGH_DIRECT with Cdb[0] set to INQUIRY (0x12)

```

The following code fragment illustrates a call to the CreateFile routine:

```

HANDLE ddHandle0, ddHandle1; // file handle for LUN0 and LUN1

/*
** Open for reading/writing on LUN0,
** where the device special file name is in the form of tapex and
** x is the logical device 0 to n - can be determined from Registry
**
** Open for media mover operations on LUN1,
** where the device special file name is in the form of
** changerx and x is the logical device 0 to n - can be determined from Registry
*/

ddHandle0 = CreateFile(
 "\\.\tape0",
 DWORD dwDesiredAccess,
 DWORD dwShareMode,
 LPSECURITY_ATTRIBUTES lpSecurityAttributes,
 DWORD dwCreationDisposition,
 DWORD dwFlagsAndAttributes,
 HANDLE hTemplateFile
);

ddHandle1 = CreateFile(
 "\\.\changer0",
 DWORD dwDesiredAccess,
 DWORD dwShareMode,

```

```

 LPSECURITY_ATTRIBUTES lpSecurityAttributes,
 DWORD dwCreationDisposition,
 DWORD dwFlagsAndAttributes,
 HANDLE hTemplateFile
);
/* Print msg if open failed for handle 0 or 1 */
if(ddHandle0 == INVALID_HANDLE_VALUE)
{
 printf("open failed for LUNn\n");
 printf("System Error = %d\n", GetLastError());
 exit (-1);
}

```

The CloseHandle entry point is called to stop I/O to the driver and device. The following code fragment illustrates a call to the CloseHandle routine:

```

BOOL rc;

rc = CloseHandle(
 ddHandle0
);
if (!rc)
{
 printf("close failed\n");
 printf("System Error = %d\n", GetLastError());
 exit (-1);
}

```

where ddHandle0 is the open file handle returned by the CreateFile call.

---

## ReadFile

The ReadFile entry point is called to read data from tape. The caller provides a buffer address and length, and the driver returns data from the tape to the buffer. The amount of data that is returned never exceeds the length parameter.

See [Variable and fixed block read/write processing](#) for a full discussion of the read/write processing feature.

The following code fragment illustrates a ReadFile call to the driver:

```

BOOL rc;

rc = ReadFile(
 HANDLE hFile,
 LPVOID lpBuffer,
 DWORD nBufferSize,
 LPDWORD lpBytesRead,
 LPOVERLAPPED lpOverlapped
);
if(rc)
{
 if (*lpBytesRead > 0)
 printf("Read %d bytes\n", *lpBytesRead);
 else
 printf("Read found file mark\n");
}
else
{
 printf("Error on read\n");
 printf("System Error = %d\n", GetLastError());
 exit (-1);
}

```

Where hFile is the open file handle, lpBuffer is the address of a buffer in which to place the data, nBufferSize is the number of bytes to be read, and lpBytesRead is the number of bytes read.

If the function succeeds, the return value rc is nonzero.

---

## WriteFile

The WriteFile entry point is called to write data to the tape. The caller provides the address and length of the buffer to be written to tape. The physical limitations of the drive can cause the write to fail. One example is attempting to write past the physical end of the tape.

See [Variable and fixed block read/write processing](#) for a full discussion of the read/write processing feature.

The following code fragment illustrates a call to the WriteFile routine:

```

BOOL rc;

rc = WriteFile(
 HANDLE hFile,
 LPCVOID lpBuffer,
 DWORD nBufferSize,
 LPDWORD lpNumberOfBytesWritten,
 LPOVERLAPPED lpOverlapped
);
if (!rc)
{
 printf("Error on write\n");
}

```

```
printf("System Error = %d\n",GetLastError());
exit (-1);
}
```

Where **hFile** is the open file handle, **lpBuffer** is the buffer address, and **nBufferSize** is the size of the buffer in bytes.

If the function succeeds, the return value rc is nonzero. The application also verifies that all the requested data was written by examining the **lpNumberOfBytesWritten** parameter. See [Write Tapemark](#) for details on committing data on the media.

## Write Tapemark

Application writers who are using the **WriteFile** entry point to write data to tape must understand that the tape device buffers data in its memory and writes that data to the media as those device buffers fill. Thus, a **WriteFile** call might return a successful return code, but the data might not be on the media yet. Calling the **WriteTapemark** entry point and receiving a good return code, however, ensures that data is committed to tape media properly if all previous **WriteFile** calls were successful. However, applications that write large amounts of data to tape might not want to wait until writing a tapemark to know whether previous data was written to the media properly. For example:

```
WriteTapemark(
 HANDLE hDevice,
 DWORD dwTapemarkType,
 DWORD dwTapemarkCount,
 BOOL bImmediate
);
```

**dwTapemarkType** is the type of operation requested.

The only type that is supported is

**TAPE\_FILEMARKS**

The **WriteTapemark** entry point might also be called with the **dwTapemarkCount** parameter set to 0 and the **bImmediate** parameter that is set to **FALSE**. This action commits any uncommitted data that is written by previous **WriteFile** calls (since the last call to **WriteTapemark**) to the media. If no error is returned by the **WriteFile** calls and the **WriteTapemark** call, the application can assume that all data is committed to the media successfully.

## SetTapePosition

The **SetTapePosition** entry point is called to seek to a particular block of media data. For example:

```
SetTapePosition(
 HANDLE hDevice,
 DWORD dwPositionMethod,
 DWORD dwPartition,
 DWORD dwOffsetLow,
 DWORD dwOffsetHigh,
 BOOL bImmediate
);
```

**dwPositionMethod** is the type of positioning.

For Magstar® devices, the following types of tapemarks and immediate values are supported.

|                     |                          |
|---------------------|--------------------------|
| TAPE_ABSOLUTE_BLOCK | bImmediate TRUE or FALSE |
| TAPE_LOGICAL_BLOCK  | bImmediate TRUE or FALSE |

For Magstar devices, there is no difference between the absolute and logical block addresses.

|                            |                          |
|----------------------------|--------------------------|
| TAPE_REWIND                | bImmediate TRUE or FALSE |
| TAPE_SPACE_END_OF_DATA     | bImmediate FALSE         |
| TAPE_SPACE_FILEMARKS       | bImmediate FALSE         |
| TAPE_SPACE_RELATIVE_BLOCKS | bImmediate FALSE         |
| TAPE_SPACE_SEQUENTIAL_FMKS |                          |

## GetTapePosition

The **GetTapePosition** entry point is called to retrieve the current tape position. For example:

```
GetTapePosition(
 HANDLE hDevice,
 DWORD dwPositionType,
 LPDWORD lpdwPartition,
 LPDWORD lpdwOffsetLow,
 LPDWORD lpdwOffsetHigh
);
```

**dwPositionType** is the type of positioning.

**TAPE\_ABSOLUTE\_POSITION** or **TAPE\_LOGICAL\_POSITION** might be specified but only the absolute position is returned.

---

## SetTapeParameters

The SetTapeParameters entry point is called to either specify the block size of a tape or set tape device data compression. The data structures are

```
struct{ // structure used by operation SET_TAPE_MEDIA_INFORMATION
 ULONG BlockSize;
}TAPE_SET_MEDIA_PARAMETERS;

struct{ // structure used by operation SET_TAPE_DRIVE_INFORMATION
 BOOLEAN ECC; // Not Supported
 BOOLEAN Compression; // Only compression can be set
 BOOLEAN DataPadding; // Not Supported
 BOOLEAN ReportSetmarks; // Not Supported
 ULONG EOTWarningZoneSize; // Not Supported
}TAPE_SET_DRIVE_PARAMETERS;

SetTapeParameters(
 HANDLE hDevice,
 DWORD dwOperation,
 LPVOID lpParameters
);
```

**dwOperation** is the type of information to set (SET\_TAPE\_MEDIA\_INFORMATION or SET\_TAPE\_DRIVE\_INFORMATION). For SET\_TAPE\_DRIVE\_INFORMATION, only compression is changeable.

**lpParameters** is the address of either a TAPE\_SET\_MEDIA\_PARAMETERS or a TAPE\_SET\_DRIVE\_PARAMETERS data structure that contains the parameters.

---

## GetTapeParameters

The GetTapeParameters entry point is called to get information that describes the tape or the tape drive.

The data structures are

```
struct{ // structure used by GET_TAPE_MEDIA_INFORMATION
 LARGE_INTEGER Capacity; /* invalid for Magstar */
 LARGE_INTEGER Remaining; /* invalid for Magstar */
 DWORD BlockSize;
 DWORD PartitionCount;
 BOOLEAN WriteProtected;
}TAPE_GET_MEDIA_PARAMETERS;

struct{ // structure used by GET_TAPE_DRIVE_INFORMATION
 BOOLEAN ECC;
 BOOLEAN Compression;
 BOOLEAN DataPadding;
 BOOLEAN ReportSetmarks;
 ULONG DefaultBlockSize;
 ULONG MaximumBlockSize;
 ULONG MinimumBlockSize;
 ULONG MaximumPartitionCount;
 ULONG FeaturesLow;
 ULONG FeaturesHigh;
 ULONG EOTWarningZoneSize;
}TAPE_GET_DRIVE_PARAMETERS;
```

The following code fragment illustrates a call to the GetTapeParameters routine.

```
DWORD rc;

rc = GetTapeParameters(
 HANDLE hDevice,
 DWORD dwOperation,
 LPDWORD lpdwSize,
 LPVOID lpParameters
);

if (rc)
{
 printf("Error on GetTapeParameters\n");
 printf("System Error = %d\n",GetLastError());
 exit (-1);
}
```

Where **hDevice** is the open file handle, **dwOperation** is the type of information requested (GET\_TAPE\_MEDIA\_INFORMATION or GET\_TAPE\_DRIVE\_INFORMATION), and **lpParameters** is the address of the returned data parameter structure.

If the function succeeds, the return value **rc** is ERROR\_SUCCESS.

---

## PrepareTape

The PrepareTape entry point is called to prepare the tape for access or removal. For example,

```
PrepareTape(
 HANDLE hDevice,
 DWORD dwOperation,
```

```

 BOOL bImmediate
);

```

**dwOperation** is the type of operation requested.

The following types of operations and immediate values are supported:

|             |                          |
|-------------|--------------------------|
| TAPE_LOAD   | bImmediate TRUE or FALSE |
| TAPE_LOCK   | bImmediate FALSE         |
| TAPE_UNLOAD | bImmediate TRUE or FALSE |
| TAPE_UNLOCK | bImmediate FALSE         |

## EraseTape

The EraseTape entry point is called to erase all or a part of a tape. The erase is completed from the current location. For example:

```

EraseTape (
 HANDLE hDevice,
 DWORD dwEraseType,
 BOOL bImmediate
);

```

**dwEraseType** is the type of operation requested.

The following types of operations and immediate values are supported.

|                 |                          |
|-----------------|--------------------------|
| TAPE_ERASE_LONG | bImmediate TRUE or FALSE |
|-----------------|--------------------------|

## GetTapeStatus

The GetTapeStatus entry point is called to determine whether the tape device is ready to process tape commands. For example,

```

GetTapeStatus (
 HANDLE hDevice
);

```

**hDevice** is the handle to the device for which to get the device status.

## DeviceIoControl

The DeviceIoControl function is described in the *Microsoft Developer Network (MSDN) Software Developer Kit(SDK)* and *Device Driver Developer Kit (DDK)*.

The DeviceIoControl function sends a control code directly to a specified device driver, causing the corresponding device to complete the specified operation.

```

BOOL DeviceIoControl (
 HANDLE hDevice, // handle to device of interest
 DWORD dwIoControlCode, // control code of operation to perform
 LPVOID lpInBuffer, // pointer to buffer to supply input data
 DWORD nInBufferSize, // size of input buffer
 LPVOID lpOutBuffer, // pointer to buffer to receive output data
 DWORD nOutBufferSize, // size of output buffer
 LPDWORD lpBytesReturned, // pointer to variable to receive output byte count
 LPOVERLAPPED lpOverlapped // pointer to overlapped structure for \
 asynchronous operation
);

```

Following is a list of the supported **dwIoControlCode** codes that are described in the MSDN DDK and used through the DeviceIoControl API.

```

IOCTL SCSI_PASS_THROUGH
 Tape and medium changer.
IOCTL SCSI_PASS_THROUGH_DIRECT
 Tape and medium changer.
IOCTL_STORAGE_RESERVE
 Tape and medium changer.
IOCTL_STORAGE_RELEASE
 Tape and medium changer.
IOCTL_STORAGE_PERSISTENT_RESERVE_IN
 Tape and medium changer.
IOCTL_STORAGE_PERSISTENT_RESERVE_OUT
 Tape and medium changer.
IOCTL_CHANGER_EXCHANGE_MEDIUM
 Medium changer not all changers.
IOCTL_CHANGER_GET_ELEMENT_STATUS
 Medium changer if bar code Reader then VolTags supported.
IOCTL_CHANGER_GET_PARAMETERS
 Medium changer.
IOCTL_CHANGER_GET_PRODUCT_DATA
 Medium changer.

```

IOCTL\_CHANGER\_GET\_STATUS  
Medium changer.  
IOCTL\_CHANGER\_INITIALIZE\_ELEMENT\_STATUS  
Medium changer with range not supported by all changers.  
IOCTL\_CHANGER\_MOVE\_MEDIUM  
Medium changer.  
IOCTL\_CHANGER\_SET\_ACCESS  
Medium changer for IE Port only and not for all changers.  
IOCTL\_CHANGER\_SET\_POSITION  
Medium changer only some devices support the transport object.

An example of the use of SCSI Pass Through is contained in the sample code SPTI.C in the DDK.

The function call DeviceIoControl is described in the SDK and examples of its use are shown in the DDK.

---

## Medium Changer IOCTLs

The Removable Storage Manager (RSM) must be stopped to use these **ioctl** commands. RSM can be stopped from Computer Management (Local) > Services and Applications > Services > Removable Storage.

- [IOCTL commands](#)

---

## IOCTL commands

Not all source or destination addresses, exchanges, moves, or operations are allowed for a particular IBM® Medium Changer. The user must issue an IOCTL\_CHANGER\_GET\_PARAMETER to determine the type of operations that are allowed by a specific changer device. Further information on allowable commands for a particular changer can be found in the IBM hardware reference for that device. It is recommended that the user have a copy of the hardware reference before any applications for the changer device are constructed.

---

### IOCTL\_CHANGER\_EXCHANGE\_MEDIUM

The media from the source element is moved to the first destination element. The medium that previously occupied the first destination element is moved to the second destination element (the second destination element might be the same as the source) by sending an **ExchangeMedium (0xA6)** SCSI command to the device. The input data is a structure of CHANGER\_EXCHANGE\_MEDIUM. This command is not supported by all devices.

---

### IOCTL\_CHANGER\_GET\_ELEMENT\_STATUS

Returns the status of all elements or of a specified number of elements of a particular type by sending a **ReadElementStatus (0xB8)** SCSI command to the device. The input and output data is a structure of CHANGER\_ELEMENT\_STATUS.

---

### IOCTL\_CHANGER\_GET\_PARAMETERS

Returns the capabilities of the changer. The output data is in a structure of GET\_CHANGER\_PARAMETERS.

---

### IOCTL\_CHANGER\_GET\_PRODUCT\_DATA

Returns the product data for the changer. The output data is in a structure of CHANGER\_PRODUCT\_DATA.

---

### IOCTL\_CHANGER\_GET\_STATUS

Returns the status of the changer by sending a **TestUnitReady (0x00)** SCSI command to the device.

---

### IOCTL\_CHANGER\_INITIALIZE\_ELEMENT\_STATUS

Initializes the status of all elements or a range of a particular element by sending an **InitializeElementStatus (0x07)** or **InitializeElementStatusWithRange (0xE7)** SCSI command to the device. The input data is a structure of CHANGER\_INITIALIZE\_ELEMENT\_STATUS.

---

### IOCTL\_CHANGER\_MOVE\_MEDIUM

Moves a piece of media from a source to a destination by sending a **MoveMedia (0xA5)** SCSI command to the device. The input data is a structure of CHANGER\_MOVE\_MEDIUM.

---

### IOCTL\_CHANGER\_REINITIALIZE\_TRANSPORT

Physically recalibrates a transport element by sending a **RezeroUnit (0x01)** SCSI command to the device. The input data is a structure of CHANGER\_ELEMENT. This command is not supported by all devices.

---

### IOCTL\_CHANGER\_SET\_ACCESS

Sets the access state of the changers IE port by sending a **PrevenAllowMediumRemoval (0x1E)** SCSI command to the device. The input data is a structure of `CHANGER_SET_ACCESS`.

## IOCTL\_CHANGER\_SET\_POSITION

Sets the changers robotic transport to a specified address by sending a **PositionToElement (0x2B)** SCSI command to the device. The input data is a structure of `CHANGER_SET_POSITION`.

## Preempt reservation

A reservation can be preempted by issuing the appropriate IOCTL. The current reservation key is needed to successfully preempt the reservation. The current reservation key can be queried by issuing an **IOCTL\_STORAGE\_PERSISTENT\_RESERVE\_IN** IOCTL:

```
PERSISTENT_RESERVE_COMMAND prcmd = { 0 };
PPRI_RESERVATION_LIST prsl = NULL;
ULONG AdditionalLength = 0;
DWORD BytesReturned = 0;
INT iStatus = 0, i, j;

UCHAR *bufDataRead = (UCHAR *) malloc(SENSE_BUFFER_SIZE * 2);

ZeroMemory(bufDataRead, SENSE_BUFFER_SIZE * 2);

prcmd.Size = sizeof(PERSISTENT_RESERVE_COMMAND);

prcmd.PR_IN.ServiceAction = RESERVATION_ACTION_READ_KEYS;
prcmd.PR_IN.AllocationLength = sizeof(PRI_REGISTRATION_LIST);

for (i = 0; i < 2; i++) {
 if (0 == i)
 AdditionalLength = sizeof(PRI_RESERVATION_LIST);

 iStatus = DeviceIoControl(tape,
 IOCTL_STORAGE_PERSISTENT_RESERVE_IN,
 &prcmd,
 prcmd.Size,
 bufDataRead,
 AdditionalLength,
 &BytesReturned,
 NULL);

 if (0 == iStatus) {
 free(bufDataRead);
 return FALSE;
 }
 prsl = (PPRI_RESERVATION_LIST)bufDataRead;

 if (0 == i) {
 AdditionalLength = (ULONG)((prsl->AdditionalLength[0] & 0xff) << 12);
 AdditionalLength |= (ULONG)((prsl->AdditionalLength[1] & 0xff) << 8);
 AdditionalLength |= (ULONG)((prsl->AdditionalLength[2] & 0xff) << 4);
 AdditionalLength |= (ULONG)(prsl->AdditionalLength[3] & 0xff);

 AdditionalLength += sizeof(PRI_RESERVATION_LIST);

 prcmd.PR_IN.AllocationLength = AdditionalLength;
 } else if (1 == i) {
 for (j = 0; (j * sizeof(PRI_RESERVATION_DESCRIPTOR)
 + sizeof(PRI_RESERVATION_LIST)) <= AdditionalLength; j++) {
 printf("\nReservation 0x%08x%08x being examined at\n",
 descriptor index %d.\n",
 ((LARGE_INTEGER*)(prsl->Reservations[j].ReservationKey))
 ->HighPart,
 ((LARGE_INTEGER*)(prsl->Reservations[j].ReservationKey))
 ->LowPart,
 j);
 }
 }
}
```

When the reservation key is known, it can be preempted, meaning a new host can be the reservation holder, thus being able to interact with the target as needed.

```
PRI_RESERVATION_DESCRIPTOR reservation = { 0 };
PERSISTENT_RESERVE_COMMAND prcmd = { 0 };
PRO_PARAMETER_LIST prolist = { 0 };
DWORD BytesReturned = 0;
INT iStatus = 0;

UCHAR bufDataRead[sizeof(PERSISTENT_RESERVE_COMMAND)
 + sizeof(PRO_PARAMETER_LIST)] = { 0 };

prcmd.Size = sizeof(PERSISTENT_RESERVE_COMMAND) + sizeof(PRO_PARAMETER_LIST);

prcmd.PR_OUT.ParameterList[1] = 0x18;
prcmd.PR_OUT.ServiceAction = 0x3;
prcmd.PR_OUT.Type = 0x3;

query_reserve(tape, &reservation);
```



```

RtlCopyMemory(prolist.ReservationKey, reservation.ReservationKey, 8);
RtlCopyMemory(prolist.ServiceActionReservationKey,
reservation.ReservationKey, 8);

RtlCopyMemory(bufDataRead, &prcmd, sizeof(PERSISTENT_RESERVE_COMMAND));
RtlCopyMemory(bufDataRead + sizeof(PERSISTENT_RESERVE_COMMAND),
&prolist, sizeof(PRO_PARAMETER_LIST));

iStatus = DeviceIoControl(tape,
 IOCTL_STORAGE_PERSISTENT_RESERVE_OUT,
 bufDataRead,
 sizeof(bufDataRead),
 bufDataRead,
 sizeof(bufDataRead),
 &BytesReturned,
 NULL);

```

The query\_reserve function can be implemented as explained earlier. Finally, if the reservation needs to be preempted on a different system than the original reservation holder, the OPEN\_ALWAYS flag comes in handy. It allows the user to query the target's serial number, then queries the reservation key, and preempts the reservation.

Caution is advised when preempting reservations due to inherent risk of data loss if done incorrectly. Applications must make sure that they are clearing or preempting the appropriate reservation.

## Vendor-specific (IBM) device IOCTLs for DeviceIoControl

The following descriptions are of the IBM® vendor-specific ioctl requests for tape and changer.

```

/*
 This macro is defined in ntddk.h and devioctl.h
 #define CTL_CODE(DeviceType, Function, Method, Access) \
 (((DeviceType) << 16) | ((Access) << 14) | ((Function) << 2) | (Method))
*/

```

The following **ioctl** commands are supported by the ibmtp.sys driver through DeviceIoControl.

```

#define LB_ACCESS FILE_READ_ACCESS | FILE_WRITE_ACCESS
#define M_MTI(x) CTL_CODE(IOCTL_BASE+2,x,METHOD_BUFFERED, LB_ACCESS)
#define IOCTL_TAPE_OBTAIN_SENSE CTL_CODE(IOCTL_TAPE_BASE, 0x0819,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_OBTAIN_VERSION CTL_CODE(IOCTL_TAPE_BASE, 0x081a,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_LOG_SELECT CTL_CODE(IOCTL_TAPE_BASE, 0x081c,
METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_TAPE_LOG_SENSE CTL_CODE(IOCTL_TAPE_BASE, 0x081d,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_LOG_SENSE10 CTL_CODE(IOCTL_TAPE_BASE, 0x0833,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_ENH_TAPE_LOG_SENSE10 CTL_CODE(IOCTL_TAPE_BASE, 0x0835, METHOD_BUFFERED,
FILE_READ_ACCESS)
#define IOCTL_TAPE_REPORT_MEDIA_DENSITY CTL_CODE(IOCTL_TAPE_BASE, 0x081e,
METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_TAPE_OBTAIN_MTDEVICE (M_MTI(16))
#define IOCTL_CREATE_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0826, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_QUERY_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0825, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_SET_ACTIVE_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0827, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_QUERY_DATA_SAFE_MODE CTL_CODE(IOCTL_TAPE_BASE, 0x0823, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_SET_DATA_SAFE_MODE CTL_CODE(IOCTL_TAPE_BASE, 0x0824, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_ALLOW_DATA_OVERWRITE CTL_CODE(IOCTL_TAPE_BASE, 0x0828, METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_SET_PEW_SIZE
CTL_CODE(IOCTL_TAPE_BASE, 0x082C, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_QUERY_PEW_SIZE
CTL_CODE(IOCTL_TAPE_BASE, 0x082B, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_VERIFY_TAPE_DATA
CTL_CODE(IOCTL_TAPE_BASE, 0x082A, METHOD_BUFFERED, FILE_READ_ACCESS)
#define IOCTL_QUERY_RAO_INFO CTL_CODE(IOCTL_TAPE_BASE, 0x082E, METHOD_BUFFERED,
FILE_READ_ACCESS)
#define IOCTL_GENERATE_RAO CTL_CODE(IOCTL_TAPE_BASE, 0x082F, METHOD_BUFFERED,
FILE_READ_ACCESS)
#define IOCTL_RECEIVE_RAO CTL_CODE(IOCTL_TAPE_BASE, 0x0834, METHOD_BUFFERED,
FILE_READ_ACCESS)

```

- [IOCTL\\_TAPE\\_OBTAIN\\_SENSE](#)
- [IOCTL\\_TAPE\\_OBTAIN\\_VERSION](#)
- [IOCTL\\_TAPE\\_LOG\\_SELECT](#)
- [IOCTL\\_TAPE\\_LOG\\_SENSE](#)
- [IOCTL\\_TAPE\\_LOG\\_SENSE10](#)
- [IOCTL\\_ENH\\_TAPE\\_LOG\\_SENSE10](#)
- [IOCTL\\_TAPE\\_REPORT\\_MEDIA\\_DENSITY](#)
- [IOCTL\\_TAPE\\_OBTAIN\\_MTDEVICE](#)
- [IOCTL\\_TAPE\\_GET\\_DENSITY](#)
- [IOCTL\\_TAPE\\_SET\\_DENSITY](#)
- [IOCTL\\_TAPE\\_GET\\_ENCRYPTION\\_STATE](#)
- [IOCTL\\_TAPE\\_SET\\_ENCRYPTION\\_STATE](#)

- [IOCTL\\_TAPE\\_SET\\_DATA\\_KEY](#)
- [IOCTL\\_CREATE\\_PARTITION](#)
- [IOCTL\\_QUERY\\_PARTITION](#)
- [IOCTL\\_SET\\_ACTIVE\\_PARTITION](#)
- [IOCTL\\_QUERY\\_DATA\\_SAFE\\_MODE](#)
- [IOCTL\\_SET\\_DATA\\_SAFE\\_MODE](#)
- [IOCTL\\_ALLOW\\_DATA\\_OVERWRITE](#)
- [IOCTL\\_READ\\_TAPE\\_POSITION](#)
- [IOCTL\\_SET\\_TAPE\\_POSITION](#)
- [IOCTL\\_QUERY\\_LBP](#)
- [IOCTL\\_SET\\_LBP](#)
- [IOCTL\\_SET\\_PEW\\_SIZE](#)
- [IOCTL\\_QUERY\\_PEW\\_SIZE](#)
- [IOCTL\\_VERIFY\\_TAPE\\_DATA](#)
- [IOCTL\\_QUERY\\_RAO\\_INFO](#)
- [IOCTL\\_GENERATE\\_RAO](#)
- [IOCTL\\_RECEIVE\\_RAO](#)
- [IOCTL\\_CHANGER\\_OBTAIN\\_SENSE](#)
- [IOCTL\\_MODE\\_SENSE](#)

---

## IOCTL\_TAPE\_OBTAIN\_SENSE

Issue this command after an error occurs to obtain sense information that is associated with the most recent error. To guarantee that the application can obtain sense information that is associated with an error, the application must issue this command before other commands to the device are issued. Subsequent operations (other than **IOCTL\_TAPE\_OBTAIN\_SENSE**) reset the sense data field before the operation is run.

This IOCTL is only available for the tape path.

The following output structure is completed by the **IOCTL\_TAPE\_OBTAIN\_SENSE** command that is passed by the caller.

```
#define MAG_SENSE_BUFFER_SIZE 96 /* Default request sense buffer size for \
 Windows 200x */

typedef struct _TAPE_OBTAIN_SENSE {
 ULONG SenseDataLength;
 // The number of bytes of valid sense data.
 // Will be zero if no error with sense data has occurred.
 // The only sense data available is that of the last error.
 CHAR SenseData[MAG_SENSE_BUFFER_SIZE];
} TAPE_OBTAIN_SENSE, *PTAPE_OBTAIN_SENSE;
```

An example of the **IOCTL\_TAPE\_OBTAIN\_SENSE** command is

```
DWORD cb;
TAPE_OBTAIN_SENSE sense_data;
DeviceIoControl(hDevice,
 IOCTL_TAPE_OBTAIN_SENSE,
 NULL,
 0,
 &sense_data,
 (long)sizeof(TAPE_OBTAIN_SENSE),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_OBTAIN\_VERSION

Issue this command to obtain the version of the device driver. It is in the form of a null terminated string.

This IOCTL is only for the tape path.

The following output structure is completed by the **IOCTL\_TAPE\_OBTAIN\_VERSION** command.

```
#define MAX_DRIVER_VERSIONID_LENGTH 12

typedef struct _TAPE_OBTAIN_VERSION {
 CHAR VersionId[MAX_DRIVER_VERSIONID_LENGTH];
} TAPE_OBTAIN_VERSION, *PTAPE_OBTAIN_VERSION;
```

An example of the **IOCTL\_TAPE\_OBTAIN\_VERSION** command is

```
DWORD cb;
TAPE_OBTAIN_VERSION code_version;
DeviceIoControl(hDevice,
 IOCTL_TAPE_OBTAIN_VERSION,
 NULL,
 0,
 &code_version,
 (long)sizeof(TAPE_OBTAIN_VERSION),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_LOG\_SELECT

This command resets all log pages that can be reset on the device to their default values. This IOCTL is only for the tape path.

An example of this command to reset all log pages follows.

```
DWORD cb;
DeviceIoControl(hDevice,
 IOCTL_TAPE_LOG_SELECT,
 NULL,
 0,
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_LOG\_SENSE

Issue this command to obtain the log data of the requested log page from IBM® Magstar® tape device. The data that is returned is formatted according to the IBM Magstar hardware reference.

This IOCTL is only for the tape path.

The following input/output structure is used by the **IOCTL\_TAPE\_LOG\_SENSE** command.

```
#define MAX_LOG_SENSE 1024 // Maximum number of bytes the command will return
typedef struct _TAPE_LOG_SENSE_PARAMETERS{
 UCHAR PageCode; // The requested log page code
 UCHAR PC; // PC = 0 for maximum values, 1 for current value, 3 for power-on values
 UCHAR PageLength[2]; /* Length of returned data, filled in by the command */
 UCHAR LogData[MAX_LOG_SENSE]; /* Log data, filled in by the command */
} TAPE_LOG_SENSE_PARAMETERS, *PTAPE_LOG_SENSE_PARAMETERS;
```

An example of the **IOCTL\_TAPE\_LOG\_SENSE** command is

```
DWORD cb;
TAPE_LOG_SENSE_PARAMETERS logsense;
logsense.PageCode=0;
logsense.PC = 1;

DeviceIoControl(hDevice,
 IOCTL_TAPE_LOG_SENSE,
 &logsense,
 (long)sizeof(TAPE_LOG_SENSE_PARAMETERS),
 &logsense,
 (long)sizeof(TAPE_LOG_SENSE_PARAMETERS),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_LOG\_SENSE10

Issue this command to obtain the log data of the requested log page/subpage from IBM® Magstar® tape device. The data returned is formatted according to the IBM Magstar hardware reference. This IOCTL is only for the tape path.

The following input/output structure is used by the **IOCTL\_TAPE\_LOG\_SENSE10** command.

```
#define MAX_LOG_SENSE 1024 // Maximum number of bytes the command will return
typedef struct _TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE{
 UCHAR PageCode; /* [IN] Log sense page */
 UCHAR SubPageCode; /* [IN] Log sense subpage */
 UCHAR PC; /* [IN] PC bit to be consistent with previous Log Sense IOCTL */
 UCHAR reserved[2]; /* unused */
 ULONG PageLength; /* [OUT] number of valid bytes in data (log_page_header_size+page_length)*/
 ULONG parm_pointer; /* [IN] specific parameter number at which the data begins */
 CHAR LogData[MAX_LOG_SENSE_DATA]; /* [OUT] log sense data */
} TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE, *PTAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE;
```

An example of the **IOCTL\_TAPE\_LOG\_SENSE10** command is

```
DWORD cb;
TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE logsense;
logsense.PageCode=0x10;
logsense.SubPageCode=0x01;
logsense.PC = 1;
DeviceIoControl(hDevice,
 IOCTL_TAPE_LOG_SENSE10,
 &logsense, (long)sizeof(TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE),
 &logsense, (long)sizeof(TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE),
 &cb, (LPOVERLAPPED) NULL);
```

---

## IOCTL\_ENH\_TAPE\_LOG\_SENSE10

Issue this command to obtain the log data of the requested log page/subpage from IBM® TotalStorage™ tape device. The data that is returned is formatted according to the IBM TotalStorage hardware reference. This IOCTL is only for the tape path and is enhanced so the application can set the page length and provide the buffer enough to get the data back. The following input/output structure is used by the **IOCTL\_ENH\_TAPE\_LOG\_SENSE10** command.

```
typedef struct _ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE{
 UCHAR PageCode; /* [IN] Log sense page */
 UCHAR SubPageCode; /* [IN] Log sense subpage */
 UCHAR PC; /* [IN] PC bit */
 UCHAR reserved[5]; /* unused */
 ULONG Length; /* [IN][OUT] number of valid bytes in data */
 /* (log_page_header_size+page_length) */
 ULONG parm_pointer; /* [IN] specific parameter number at which */
 /* the data begins */
 CHAR LogData[1]; /* [IN] log sense buffer allocated by */
 /* application */
 /* [OUT] log sense data */
} ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE, *PENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE;
```

An example of the **IOCTL\_ENH\_TAPE\_LOG\_SENSE10** command is

```
DWORD cb;
char *logsense;
int pageLength = 256;
long lsize = sizeof(ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE) - sizeof
(CHAR) /*LogData[1]*/ + pageLength

logsense = malloc (lsize);
(ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE)logsense->PageCode=0x10;
(ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE)logsense->SubPageCode=0x01;
(ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE)logsense->PC = 1;
(ENH_TAPE_LOG_SENSE_PARAMETERS_WITH_SUBPAGE)logsense->Length = pageLength;
DeviceIoControl (hDevice,
 IOCTL_ENH_TAPE_LOG_SENSE10,
 &logsense, (long)lsize,
 &logsense, (long)lsize,
 &cb, (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_REPORT\_MEDIA\_DENSITY

Issue this command to obtain the media density information on the loaded media in the drive. If there is no media load, the command fails. This IOCTL is only for the tape path.

The following output structure is completed by the **IOCTL\_TAPE\_REPORT\_MEDIA\_DENSITY** command.

```
typedef struct TAPE_REPORT_DENSITY{
 ULONG PrimaryDensityCode; /* Primary Density Code */
 ULONG SecondaryDensityCode; /* Secondary Density Code */
 BOOLEAN WriteOk; /* 0 = does not support writing in this format */
 /* 1 = support writing in this format */
 ULONG BitsPerMM; /* Bits Per mm */
 ULONG MediaWidth; /* Media Width */
 ULONG Tracks; /* Tracks */
 ULONG Capacity; /* Capacity in MegaBytes */
} TAPE_REPORT_DENSITY, *PTAPE_REPORT_DENSITY;
```

An example of the **IOCTL\_TAPE\_REPORT\_MEDIA\_DENSITY** command is

```
DWORD cb;
TAPE_REPORT_DENSITY tape_reportden;

DeviceIoControl (hDevice,
 IOCTL_TAPE_REPORT_MEDIA_DENSITY,
 NULL,
 0,
 &tape_reportden,
 (long)sizeof(TAPE_REPORT_DENSITY),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_OBTAIN\_MTDEVICE

Issue this command to obtain the device number of a 3590 TotalStorage™ device in an IBM® 3494 Enterprise Tape Library. An error is returned if it is issued against a 3570 drive.

The following output structure is filled in by the **IOCTL\_TAPE\_OBTAIN\_MTDEVICE** command.

```
typedef ULONG TAPE_OBTAIN_MTDEVICE, *PTAPE_OBTAIN_MTDEVICE;
```

An example of the **IOCTL\_TAPE\_OBTAIN\_MTDEVICE** command is

```
int *rc_ptr
DWORD cb;
```

```
TAPE_OBTAIN_MTDEVICE mt_device;

*rc_ptr = DeviceIoControl(gp->ddHandle0,
 IOCTL_TAPE_OBTAIN_MTDEVICE,
 NULL,
 0,
 &mt_device,
 (long)sizeof(TAPE_OBTAIN_MTDEVICE),
 &cb,
 (LPOVERLAPPED) NULL);

if(*rc_ptr)
 printf(fp, "\nnntutil MTDevice Info : %x\n\n", mt_device);
else
 /* Error handling code */
```

---

## IOCTL\_TAPE\_GET\_DENSITY

The IOCTL code for **IOCTL\_TAPE\_GET\_DENSITY** is defined as follows.

```
#define IOCTL_TAPE_GET_DENSITY \
CTL_CODE(IOCTL_TAPE_BASE, 0x000c, METHOD_BUFFERED, \
 FILE_READ_ACCESS | FILE_WRITE_ACCESS).
```

The IOCTL reports density for supported devices by using the following structure.

```
typedef struct _TAPE_DENSITY
{
 UCHAR ucDensityCode;
 UCHAR ucDefaultDensity;
 UCHAR ucPendingDensity;
} TAPE_DENSITY, *PTAPE_DENSITY;
```

An example of the **IOCTL\_TAPE\_GET\_DENSITY** command is

```
TAPE_DENSITY tape_density = {0};
```

```
rc = DeviceIoControl(hDevice,
 IOCTL_TAPE_GET_DENSITY,
 NULL,
 0,
 &tape_density,
 sizeof(TAPE_DENSITY),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_SET\_DENSITY

The IOCTL code for **IOCTL\_TAPE\_SET\_DENSITY** is defined as follows.

```
#define IOCTL_TAPE_SET_DENSITY \
CTL_CODE(IOCTL_TAPE_BASE, 0x000d, METHOD_BUFFERED, \
 FILE_READ_ACCESS | FILE_WRITE_ACCESS)
```

The IOCTL sets density for supported devices by using the following structure.

```
typedef struct _TAPE_DENSITY
{
 UCHAR ucDensityCode;
 UCHAR ucDefaultDensity;
 UCHAR ucPendingDensity;
} TAPE_DENSITY, *PTAPE_DENSITY;
```

**ucDensityCode** is ignored. **ucDefaultDensity** and **ucPendingDensity** are set by using the tape drive's mode page 0x25. Caution must be taken when this IOCTL is issued. An incorrect tape density might lead to data corruption.

An example of the **IOCTL\_TAPE\_SET\_DENSITY** command is

```
TAPE_DENSITY tape_density;

// Modify fields of tape_density. For details, see the SCSI specification
// for your hardware.

rc = DeviceIoControl(hDevice,
 IOCTL_TAPE_SET_DENSITY,
 &tape_density,
 sizeof(TAPE_DENSITY),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_GET\_ENCRYPTION\_STATE

This IOCTL command queries the drive's encryption method and state.

The IOCTL code for **IOCTL\_TAPE\_GET\_ENCRYPTION\_STATE** is defined as follows.

```
#define IOCTL_TAPE_GET_ENCRYPTION_STATE CTL_CODE(IOCTL_TAPE_BASE, 0x0820,
METHOD_BUFFERED, FILE_READ_ACCESS)
```

The IOCTL gets encryption states for supported devices by using the following structure.

```
typedef struct _ENCRYPTION_STATUS
{
 UCHAR ucEncryptionCapable; /* (1) Set this field as a boolean based on
 the capability of the drive */
 UCHAR ucEncryptionMethod; /* (2) Set this field to one of the
 defines METHOD_* below */
 UCHAR ucEncryptionState; /* (3) Set this field to one of the
 defines STATE_* below */
 UCHAR aucReserved[13];
} ENCRYPTION_STATUS, *PENCRYPTION_STATUS;

#defines for METHOD.

#define ENCRYPTION_METHOD_NONE 0 /* Only used in
 GET_ENCRYPTION_STATE */
#define ENCRYPTION_METHOD_LIBRARY 1 /* Only used in
 GET_ENCRYPTION_STATE */
#define ENCRYPTION_METHOD_SYSTEM 2 /* Only used in
 GET_ENCRYPTION_STATE */
#define ENCRYPTION_METHOD_APPLICATION 3 /* Only used in
 GET_ENCRYPTION_STATE */
#define ENCRYPTION_METHOD_CUSTOM 4 /* Only used in
 GET_ENCRYPTION_STATE */
#define ENCRYPTION_METHOD_UNKNOWN 5 /* Only used in
 GET_ENCRYPTION_STATE */

#defines for STATE.

#define ENCRYPTION_STATE_OFF 0 /* Used in GET/SET_ENCRYPTION_STATE */
#define ENCRYPTION_STATE_ON 1 /* Used in GET/SET_ENCRYPTION_STATE */
#define ENCRYPTION_STATE_NA 2 /* Only used in GET_ENCRYPTION_STATE */
```

An example of the **IOCTL\_TAPE\_GET\_ENCRYPTION\_STATE** command is

```
ENCRYPTION_STATUS scEncryptStat;
DeviceIoControl(hDevice,
 IOCTL_TAPE_GET_ENCRYPTION_STATE,
 &scEncryptStat,
 sizeof(ENCRYPTION_STATUS),
 &scEncryptStat,
 sizeof(ENCRYPTION_STATUS),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_SET\_ENCRYPTION\_STATE

This IOCTL command allows only set encryption state for application-managed encryption.

Note: On unload, some drive settings might be reset to default. To set the encryption state, the application must issue this IOCTL after a tape is loaded and at BOP. The data structure that is used for this IOCTL is the same as for **IOCTL\_GET\_ENCRYPTION\_STATE**.

```
#define IOCTL_TAPE_SET_ENCRYPTION_STATE CTL_CODE(IOCTL_TAPE_BASE, 0x0821,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
```

An example of the **IOCTL\_TAPE\_SET\_ENCRYPTION\_STATE** command is

```
ENCRYPTION_STATUS scEncryptStat;
DeviceIoControl(hDevice,
 IOCTL_TAPE_SET_ENCRYPTION_STATE,
 &scEncryptStat,
 sizeof(ENCRYPTION_STATUS),
 &scEncryptStat,
 sizeof(ENCRYPTION_STATUS),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_TAPE\_SET\_DATA\_KEY

This IOCTL command is used to set the data key only for application-managed encryption.

The IOCTL sets data keys for supported devices by using the following structure.

```
#define IOCTL_TAPE_SET_DATA_KEY CTL_CODE(IOCTL_TAPE_BASE, 0x0822,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)

#define DATA_KEY_INDEX_LENGTH 12
#define DATA_KEY_RESERVED1_LENGTH 15
```

```
#define DATA_KEY_LENGTH 32
#define DATA_KEY_RESERVED2_LENGTH 48
typedef struct _DATA_KEY
{
 UCHAR aucDataKeyIndex[DATA_KEY_INDEX_LENGTH];
 UCHAR ucDataKeyIndexLength;
 UCHAR aucReserved1[DATA_KEY_RESERVED1_LENGTH];
 UCHAR aucDataKey[DATA_KEY_LENGTH];
 UCHAR aucReserved2[DATA_KEY_RESERVED2_LENGTH];
} DATA_KEY, *PDATA_KEY;
```

An example of the **IOCTL\_TAPE\_SET\_DATA\_KEY** command is

```
DATA_KEY scDataKey;
/* fill in your data key and data key length, then issue DeviceIoControl */
DeviceIoControl(hDevice,
 IOCTL_TAPE_SET_DATA_KEY,
 &scDataKey,
 sizeof(DATA_KEY),
 &scDataKey,
 sizeof(DATA_KEY),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_CREATE\_PARTITION

This command is used to create one or more partitions on the tape. The tape must be at BOT (partition 0 logical block id 0) before the command is issued or it fails. The application must either issue this **IOCTL\_CREATE\_PARTITION** after a tape is initially loaded or issue the **IOCTL\_SET\_ACTIVE\_PARTITION** with the partition\_number and logical\_clock\_id fields that are set to 0 first.

The structure that is used to create partitions is

```
#define IOCTL_CREATE_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0826,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
typedef struct _TAPE_PARTITION{
 UCHAR type; /* Type of tape partition to create */
 UCHAR number_of_partitions; /* Number of partitions to create */
 UCHAR size_unit; /* IDP size unit of partition sizes below */
 USHORT size[MAX_PARTITIONS]; /* Array of partition sizes in size units */
 /* for each partition, 0 to (number - 1) */
 /* Size can not be 0 and one partition */
 /* size must be 0xFFFF to use the */
 /* remaining capacity on the tape. */
 UCHAR partition_method; /* partitioning type */
 char reserved [31];
} TAPE_PARTITION, *PTAPE_PARTITION;
```

An example of the **IOCTL\_CREATE\_PARTITION** command is

```
DWORD cb;
TAPE_PARTITION tape_partition
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_CREATE_PARTITION,
 &tape_partition,
 (long)sizeof(TAPE_PARTITION),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_QUERY\_PARTITION

This command returns partition information for the current loaded tape.

The following output structure is completed by the **IOCTL\_QUERY\_PARTITION** command.

```
#define IOCTL_QUERY_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0825,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define MAX_PARTITIONS 255
typedef struct _QUERY_PARTITION{
 UCHAR max_partitions; /* Max number of supported partitions */
 UCHAR active_partition; /* current active partition on tape */
 UCHAR number_of_partitions; /* Number of partitions from 1 to max */
 UCHAR size_unit; /* Size unit of partition sizes below */
 USHORT size[MAX_PARTITIONS]; /* Array of partition sizes in size units */
 /* for each partition, 0 to (number - 1) */
 UCHAR partition_method; /* partitioning type */
 char reserved [31];
} QUERY_PARTITION, *PQUERY_PARTITION;
```

An example of the **IOCTL\_QUERY\_PARTITION** command is

```
DWORD cb;
QUERY_PARTITION tape_query_partition;
```

```
DeviceIoControl(gp->ddHandle0,
 IOCTL_QUERY_PARTITION,
 NULL,
 0,
 &tape_query_partition,
 (long)sizeof(QUERY_PARTITION),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_SET\_ACTIVE\_PARTITION

This command is used to set the current active partition that is used on tape and locate to a specific logical block id within the partition. If the logical block id is 0, the tape is positioned at BOP. If the partition number specified is 0 along with a logical block id 0, the tape is positioned at both BOP and BOT.

The structure for **IOCTL\_SET\_ACTIVE\_PARTITION** command is

```
#define IOCTL_SET_ACTIVE_PARTITION CTL_CODE(IOCTL_TAPE_BASE, 0x0827,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
typedef struct _SET_ACTIVE_PARTITION{
 UCHAR partition_number; /* Partition number 0-n to change to */
 ULONGLONG logical_block_id; /* Blockid to locate to within partition */
 char reserved[32];
} SET_ACTIVE_PARTITION, *PSET_ACTIVE_PARTITION;
```

An example of the **IOCTL\_SET\_ACTIVE\_PARTITION** command is

```
DWORD cb;
SET_ACTIVE_PARTITION set_partition;
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_SET_ACTIVE_PARTITION,
 &set_partition,
 (long)sizeof(SET_ACTIVE_PARTITION),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_QUERY\_DATA\_SAFE\_MODE

This command reports if the Data Safe Mode is enabled or disabled.

The following output structure is completed by the **IOCTL\_QUERY\_DATA\_SAFE\_MODE** command.

```
#define IOCTL_QUERY_DATA_SAFE_MODE CTL_CODE(IOCTL_TAPE_BASE, 0x0823,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
typedef struct _DATA_SAFE_MODE{
 ULONG value;
} DATA_SAFE_MODE, *PDATA_SAFE_MODE;
```

An example of the **IOCTL\_QUERY\_DATA\_SAFE\_MODE** command is

```
DWORD cb;
DATA_SAFE_MODE tapeDataSafeMode;
DeviceIoControl(gp->ddHandle0,
 IOCTL_QUERY_DATA_SAFE_MODE,
 NULL,
 0,
 &tapeDataSafeMode,
 (long)sizeof(DATA_SAFE_MODE),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_SET\_DATA\_SAFE\_MODE

This command enables or disables Data Safe Mode.

The structure that is used to enable or disable Data Safe Mode is the same as **IOCTL\_QUERY\_DATA\_SAFE\_MODE**.

An example of the **IOCTL\_SET\_DATA\_SAFE\_MODE** command is

```
#define IOCTL_SET_DATA_SAFE_MODE CTL_CODE(IOCTL_TAPE_BASE, 0x0824,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
DATA_SAFE_MODE tapeDataSafeMode;
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_SET_DATA_SAFE_MODE,
 &tapeDataSafeMode,
 (long)sizeof(DATA_SAFE_MODE),
 NULL,
```



```
0,
&cb,
(LPOVERLAPPED) NULL);
```

## IOCTL\_ALLOW\_DATA\_OVERWRITE

This command allows previously written data on the tape to be overwritten. This action happens when append only mode is enabled on the drive with either a write type command or a format command is allowed on the **IOCTL\_CREATE\_PARTITION**. Before this IOCTL is issued, the application must locate to the partition number and logical block id within the partition where the data overwrite or format occurs.

The data structure that is used for **IOCTL\_ALLOW\_DATA\_OVERWRITE** to enable or disable is

```
#define IOCTL_ALLOW_DATA_OVERWRITE CTL_CODE(IOCTL_TAPE_BASE, 0x0828,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
typedef struct ALLOW_DATA_OVERWRITE{
 UCHAR partition_number; /* Partition number 0-n to overwrite */
 ULONGULONG logical_block_id; /* Blockid to overwrite to within partition */
 UCHAR allow_format_overwrite; /* allow format if in data safe mode */
 UCHAR reserved[32];
} ALLOW_DATA_OVERWRITE, *PALLOW_DATA_OVERWRITE;
```

An example of the **IOCTL\_ALLOW\_DATA\_OVERWRITE** command is

```
ALLOW_DATA_OVERWRITE tapeAllowDataOverwrite;
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_ALLOW_DATA_OVERWRITE,
 &tapeAllowDataOverwrite,
 (long)sizeof(ALLOW_DATA_OVERWRITE),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);
```

## IOCTL\_READ\_TAPE\_POSITION

This command returns Position data in either the short, long, or extended form. The type of data to return is specified by setting the data\_format field to either RP\_SHORT\_FORM, RP\_LONG\_FORM, or RP\_EXTENDED\_FORM.

The data structures that are used with this IOCTL are

```
#define IOCTL_READ_TAPE_POSITION CTL_CODE(IOCTL_TAPE_BASE, 0x0829,
METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define RP_SHORT_FORM 0x00
#define RP_LONG_FORM 0x06
#define RP_EXTENDED_FORM 0x08

typedef struct _SHORT_DATA_FORMAT {
 UCHAR bop:1, /* beginning of partition */
 eop:1, /* end of partition */
 locu:1, /* 1 means num_buffer_logical_obj field is unknown */
 bycu:1, /* 1 means the num_buffer_bytes field is unknown */
 rsvd:1,
 lolu:1, /* 1 means the first and last logical obj position fields
 are unknown */
 perr:1, /* 1 means the position fields have overflowed and
 cannot be reported */
 bpew:1, /* beyond programmable early warning */
 UCHAR active_partition; /* current active partition */
 UCHAR reserved[2];
 UCHAR first_logical_obj_position[4]; /* current logical object position */
 UCHAR last_logical_obj_position[4]; /* next logical object to be
 transferred to tape */
 UCHAR num_buffer_logical_obj[4]; /* number of logical objects in buffer */
 UCHAR num_buffer_bytes[4]; /* number of bytes in buffer */
 UCHAR reserved1; /* instead of the commented reserved1 */
} SHORT_DATA_FORMAT, *PSHORT_DATA_FORMAT;

typedef struct _LONG_DATA_FORMAT {
 UCHAR bop:1, /* beginning of partition */
 eop:1, /* end of partition */
 rsvd1:2,
 mpu:1, /* 1 means the logical file id field is unknown */
 lonu:1, /* 1 means either the partition number or logical obj number field
 are unknown */
 rsvd2:1,
 bpew:1, /* beyond programmable early warning */
 CHAR reserved[6];
 UCHAR active_partition; /* current active partition */
 UCHAR logical_obj_number[8]; /* current logical object position */
 UCHAR logical_file_id[8]; /* number of filemarks from bop and
 current logical position */
 UCHAR obsolete[8];
} LONG_DATA_FORMAT, *PLONG_DATA_FORMAT;

typedef struct _EXTENDED_DATA_FORMAT {
```

```

 UCHAR bop:1, /* beginning of partition */
 eop:1, /* end of partition */
 locu:1, /* 1 means num_buffer_logical_obj field is unknown */
 bycu:1, /* 1 means the num_buffer_bytes field is unknown */
 rsvd :1,
 lolu:1, /* 1 means the first and last logical obj position fields
 are unknown */
 perr: 1, /* 1 means the position fields have overflowed and
 can not be reported */
 bpew :1, /* beyond programmable early warning */
 UCHAR active_partition; /* current active partition */
 UCHAR additional_length[2];
 UCHAR num_buffer_logical_obj[4]; /* number of logical objects in buffer */
 UCHAR first_logical_obj_position[8]; /* current logical object position */
 UCHAR last_logical_obj_position[8]; /* next logical object to be
 transferred to tape */
 UCHAR num_buffer_bytes[8]; /* number of bytes in buffer */
 UCHAR reserved;
} EXTENDED_DATA_FORMAT, *PEXTENDED_DATA_FORMAT;

typedef struct READ_TAPE_POSITION{
 UCHAR data_format; /* Specifies the return data format either
 short, long or extended*/
 union
 {
 SHORT_DATA_FORMAT rp_short;
 LONG_DATA_FORMAT rp_long;
 EXTENDED_DATA_FORMAT rp_extended;
 UCHAR reserved[64];
 } rp_data;
} READ_TAPE_POSITION, *PREAD_TAPE_POSITION;

```

An example of the **READ\_TAPE\_POSITION** command is

```

DWORD cb;
READ_TAPE_POSITION tapePosition;
*rc_ptr = DeviceIoControl(gp->ddHandle0,
 IOCTL_READ_TAPE_POSITION,
 &tapePosition,
 (long)sizeof(READ_TAPE_POSITION),
 &tapePosition,
 (long)sizeof(READ_TAPE_POSITION),
 &cb,
 (LPOVERLAPPED) NULL);

```

---

## IOCTL\_SET\_TAPE\_POSITION

This command is used to position the tape in the current active partition to either a logical block id or logical filemark. The logical\_id\_type field in the IOCTL structure specifies either a logical block or logical filemark.

The data structure that is used with this IOCTL is

```

#define IOCTL_SET_TAPE_POSITION_LOCATE16 CTL_CODE(IOCTL_TAPE_BASE, 0x0830,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define LOGICAL_ID_BLOCK_TYPE 0x00
#define LOGICAL_ID_FILE_TYPE 0x01

typedef struct _SET_TAPE_POSITION{
 UCHAR logical_id_type; /* Block or file as defined above */
 ULONGLONG logical_id; /* logical object or logical file to position to */
 UCHAR reserved[32];
} SET_TAPE_POSITION, *PSET_TAPE_POSITION;

```

An example of the **SET\_TAPE\_POSITION** command is

```

DWORD cb;
SET_TAPE_POSITION tapePosition;

*rc_ptr = DeviceIoControl(gp->ddHandle0,
 IOCTL_SET_TAPE_POSITION_LOCATE16,
 &tapePosition,
 (long)sizeof(SET_TAPE_POSITION)
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);

```

---

## IOCTL\_QUERY\_LBP

This command returns logical block protection information. The following output structure is completed by the **IOCTL\_QUERY\_LBP** command.

```

#define IOCTL_QUERY_LBP CTL_CODE(IOCTL_TAPE_BASE, 0x0831,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
typedef struct _LOGICAL_BLOCK_PROTECTION {
 UCHAR lbp_capable; /* [OUTPUT] the capability of lbp for QUERY ioctl only */

```

```

 UCHAR lbp_method; /* lbp method used for QUERY [OUTPUT] and SET [INPUT] ioctls */
#define LBP_DISABLE 0x00
#define REED_SOLOMON_CRC 0x01
 UCHAR lbp_info_length; /* lbp info length for QUERY [OUTPUT] and SET [INPUT]
 ioctls */
 UCHAR lbp_w; /* protection info included in write data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 UCHAR lbp_r; /* protection info included in read data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 UCHAR rbdp; /* protection info included in recover buffer data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 UCHAR reserved[26];
}LOGICAL_BLOCK_PROTECTION, *PLOGICAL_BLOCK_PROTECTION;

```

An example of the **IOCTL\_QUERY\_LBP** command is

```

*rc_ptr = DeviceIoControl(gp->ddHandle0,
 IOCTL_QUERY_LBP,
 NULL,
 0,
 &tape_query_LBP,
 (long)sizeof(LOGICAL_BLOCK_PROTECTION),
 &cb,
 (LPOVERLAPPED) NULL);

```

---

## IOCTL\_SET\_LBP

This command sets logical block protection information. The following input structure is sent to the **IOCTL\_SET\_LBP** command.

```

#define IOCTL_SET_LBP CTL_CODE(IOCTL_TAPE_BASE, 0x0832,
METHOD_BUFFERED,
FILE_READ_ACCESS | FILE_WRITE_ACCESS)
typedef struct _LOGICAL_BLOCK_PROTECTION {
 UCHAR lbp_capable; /* [OUTPUT] the capability of lbp for QUERY ioctl only */
 UCHAR lbp_method; /* lbp method used for QUERY [OUTPUT] and SET [INPUT]
 ioctls */
#define LBP_DISABLE 0x00
#define REED_SOLOMON_CRC 0x01
 UCHAR lbp_info_length; /* lbp info length for QUERY [OUTPUT] and SET [INPUT]
 ioctls */
 UCHAR lbp_w; /* protection info included in write data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 UCHAR lbp_r; /* protection info included in read data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 UCHAR rbdp; /* protection info included in recover buffer data */
 /* a boolean for QUERY [OUTPUT] and SET [INPUT] ioctls */
 UCHAR reserved[26];
}LOGICAL_BLOCK_PROTECTION, *PLOGICAL_BLOCK_PROTECTION;

```

An example of the **IOCTL\_SET\_LBP** command is

```

*rc_ptr = DeviceIoControl(gp->ddHandle0,
 IOCTL_SET_LBP,
 &tape_set_LBP,
 (long)sizeof(LOGICAL_BLOCK_PROTECTION),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);

```

---

## IOCTL\_SET\_PEW\_SIZE

This command is used to set Programmable Early Warning size.

```

#define IOCTL_SET_PEW_SIZE
CTL_CODE(IOCTL_TAPE_BASE, 0x082C, METHOD_BUFFERED, FILE_READ_ACCESS)

```

The structure that is used to set PEW size is

```

typedef struct _PEW_SIZE{
 USHORT value;
} PEW_SIZE, *PPEW_SIZE;

```

An example of the **IOCTL\_SET\_PEW\_SIZE** command is

```

DWORD cb;
PEW_SIZE pew_size;
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_SET_PEW_SIZE,
 &pew_size, (long)sizeof(PEW_SIZE),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);

```

---

## IOCTL\_QUERY\_PEW\_SIZE

This command is used to query Programmable Early Warning size.

```
#define IOCTL_QUERY_PEW_SIZE
CTL_CODE(IOCTL_TAPE_BASE, 0x082B, METHOD_BUFFERED, FILE_READ_ACCESS)
```

The structure that is used to query PEW size is

```
typedef struct _PEW_SIZE{
 USHORT value;
} PEW_SIZE, *PPEW_SIZE;
```

An example of the **IOCTL\_QUERY\_PEW\_SIZE** command is

```
DWORD cb;
PEW_SIZE pew_size;
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_QUERY_PEW_SIZE,
 NULL,
 0,
 &pew_size,
 (long)sizeof(PEW_SIZE),
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_VERIFY\_TAPE\_DATA

This command is used to verify tape data. It uses the drive's error detection and correction hardware to determine whether it can be recovered from the tape. It also checks whether the protection information is present and validates correctly on logical block on the medium. It returns a failure or a success.

```
#define IOCTL_VERIFY_TAPE_DATA
CTL_CODE(IOCTL_TAPE_BASE, 0x082A, METHOD_BUFFERED, FILE_READ_ACCESS)
```

The structure that is used to verify tape data is

```
typedef struct _VERIFY_DATA {
 UCHAR reserved : 2; /* Reserved */
 UCHAR vte: 1; /* [IN] verify to end-of-data */
 UCHAR vlbp: 1; /* [IN] verify logical block protection information */
 UCHAR vbf: 1; /* [IN] verify by filemarks */
 UCHAR immed: 1; /* [IN] return SCSI status immediately */
 UCHAR bytcmp: 1; /* No use currently */
 UCHAR fixed: 1; /* [IN] set Fixed bit to verify the length of
 each logical block */
 UCHAR reserved[15];
 ULONG verify_length; /* [IN] amount of data to be verified */
} VERIFY_DATA, *PVERIFY_DATA;
```

An example of the **IOCTL\_VERIFY\_DATA** command is

```
DWORD cb;
VERIFY_DATA verify_data;
...
DeviceIoControl(gp->ddHandle0,
 IOCTL_VERIFY_TAPE_DATA,
 &verify_data,
 sizeof(VERIFY_DATA),
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);
```

---

## IOCTL\_QUERY\_RAO\_INFO

This command is used to query the maximum UDS number and UDS size before the Recommended Access Order list is generated and received (TS1140 or later). The structure for the **IOCTL\_QUERY\_RAO\_INFO** command is

```
#define UDS_WITHOUT_GEOMETRY 0
#define UDS_WITH_GEOMETRY 1

typedef struct _QUERY_RAO_INF{
 CHAR uds_type; /* [IN] 0: UDS_WITHOUT_GEOMETRY
 1: UDS_WITH_GEOMETRY */
 CHAR reserved[7]; USHORT max_uds_number; /* [OUT] Max UDS number supported
 from drive */
 USHORT max_uds_size; /* [OUT] Max single UDS size supported
 from drive in bytes */
 USHORT max_host_uds_number; /* [OUT] Max UDS number supported from
 driver */
} QUERY_RAO_INFO, *PQUERY_RAO_INFO;
```

An example of the **IOCTL\_QUERY\_RAO\_INFO** command is

```

QUERY_RAO_INFO qRAO;
...
qRAO.uds_type = udstype; //UDS_WITHOUT_GEOMETRY or UDS_WITH_GEOMETRY
*rc_ptr = DeviceIoControl(hDevice,
 IOCTL_QUERY_RAO_INFO, &
 qRAO,
 sizeof(QUERY_RAO_INFO),
 &qRAO,
 sizeof(QUERY_RAO_INFO),
 &cb,
 (LPOVERLAPPED) NULL);

```

## IOCTL\_GENERATE\_RAO

This command is used to generate a Recommended Access Order list (TS1140 or later). The UDS list is required as input. Use **USD\_DESCRIPTOR** to build this list. The structure for the **IOCTL\_GENERATE\_RAO** command is

```

#define UDS_WITHOUT_GEOMETRY 0
#define UDS_WITH_GEOMETRY 1

typedef struct _GENERATE_RAO{
 CHAR process; /* [IN] Requested process to generate RAO list */
 /* 0: no reorder UDS and no calculate locate time */
 /* (not currently supported by the drive) */
 /* 1: no reorder UDS but calculate locate time */
 /* 2: reorder UDS and calculate locate time */
 CHAR uds_type; /* [IN] 0: UDS_WITHOUT_GEOMETRY */
 /* 1: UDS_WITH_GEOMETRY */
 CHAR reserved1[2];
 ULONG grao_list_length; /* [IN] The data length is allocated for GRAO list. */
 CHAR reserved2[8];
 CHAR grao_list[1]; /* [IN] the pointer is allocated to the size */
 /* of grao_list_leng */
 /* (uds_number * sizeof(struct grao_uds_desc) */
 /* +sizeof(struct grao_list_header)) */
 /* and contains the data of GRAO parameter list. */
 /* The uds number isn't larger */
 /* than max_host_uds_number in QUERY_RAO ioctl. */
} GENERATE_RAO, *PGENERATE_RAO;

typedef struct _UDS_DESCRIPTOR{
 CHAR descLength[2];
 CHAR reserved[3];
 CHAR UDSName[10];
 CHAR PartNum;
 CHAR beginningLOI[8];
 CHAR endingLOI[8];
}UDS_DESCRIPTOR, *PUDS_DESCRIPTOR;

```

An example of the **IOCTL\_GENERATE\_RAO** command is

```

UDS_SIZE 32
HEADER_SIZE 8

char *pGRAO;
...
long lGRAOsize = sizeof(GENERATE_RAO)-sizeof(CHAR)/*grao_list[1]
*/ + udsamount*UDS_SIZE + HEADER_SIZE;
pGRAO = malloc (lGRAOsize);
...
((PGRAO)pGRAO)->process = process;
((PGRAO)pGRAO)->uds_type = udstype;
((PGRAO)pGRAO)->grao_list_length = udsamount*UDS_SIZE + HEADER_SIZE;
...
((PGRAO_LIST_HEADER)((PGRAO)pGRAO)->grao_list)->addl_data = ((PGRAO)pGRAO)->
grao_list_length - HEADER_SIZE;
PopulateUDS ((PGRAO_LIST_HEADER)((PGRAO)pGRAO)->grao_list)+HEADER_SIZE, udsamount);

*rc_ptr = DeviceIoControl(hDevice,
 IOCTL_GENERATE_RAO,
 pGRAO,
 lGRAOsize,
 NULL,
 0,
 &cb,
 (LPOVERLAPPED) NULL);

```

## IOCTL\_RECEIVE\_RAO

This command is used to receive the generated Recommended Access Order list (TS1140 and later).

The structure for **IOCTL\_RECEIVE\_RAO** command is

```

typedef struct _RECEIVE_RAO_LIST {
 ULONG rrao_list_offset; /* [IN] The offset of receive RAO list to
 /* begin returning data */
}

```

```

 ULONG rrao_list_length; /* [IN/OUT] number byte of data length */
 /* [IN] The data length is allocated for RRAO list */
 /* by application */
 /* the length is (max_uds_size * uds_number + sizeof */
 /* (struct rrao_list_header) */
 /* max_uds_size is reported in QUERY_RAO_INFO ioctl */
 /* uds_number is the total UDS number requested */
 /* from application */
 /* in GENERATE_RAO ioctl */
 /* [OUT] the data length is actual returned in RRAO list */
 /* from the driver */
 CHAR reserved[8];
 CHAR rrao_list[1]; /* [IN/OUT] the data pointer of RRAO list */
}RECEIVE_RAO_LIST, *PRECEIVE_RAO_LIST;

```

An example of the **IOCTL\_RECEIVE\_RAO** command is

```

HEADER_SIZE 8
...
char pRRAO;
...
long lRRAOsize=sizeof(RECEIVE_RAO_LIST)-sizeof(CHAR)/*rrao_list[1]
*/ + udsamount*udssize+UDS_HEADER;
...
pRRAO = malloc (lRRAOsize);
...
((PRECEIVE_RAO_LIST) pRRAO)->rrao_list_offset = offset;
((PRECEIVE_RAO_LIST) pRRAO)->rrao_list_length = udsamount*udssize+UDS_HEADER;
...
*rc_ptr = DeviceIoControl(hDevice,
 IOCTL_RECEIVE_RAO,
 pRRAO,
 lRRAOsize,
 pRRAO,
 lRRAOsize,
 &cb,
 (LPOVERLAPPED) NULL);

```

## IOCTL\_CHANGER\_OBTAIN\_SENSE

Issue this command after an error occurs to obtain sense information that is associated with the most recent error. To guarantee that the application can obtain sense information that is associated with an error, the application must issue this command before other commands are issued to the device. Subsequent operations (other than **IOCTL\_CHANGER\_OBTAIN\_SENSE**) reset the sense data field before the operation is run.

This IOCTL is only available for the changer path.

```

#define IOCTL_CHANGER_BASE FILE_DEVICE_CHANGER
#define IOCTL_CHANGER_OBTAIN_SENSE
 CTL_CODE(IOCTL_CHANGER_BASE, 0x0819, METHOD_BUFFERED, FILE_READ_ACCESS)

```

The following output structure is completed by the **IOCTL\_CHANGER\_OBTAIN\_SENSE** command that is passed by the caller.

```

#define MAG_SENSE_BUFFER_SIZE 96 /* Default request sense buffer size for \
Windows 200x */
typedef struct _CHANGER_OBTAIN_SENSE {
 ULONG SenseDataLength; /* The number of bytes of valid sense data.
 // Will be zero if no error with sense data has occurred.
 // The only sense data available is that of the last error.
 CHAR SenseData[MAG_SENSE_BUFFER_SIZE];
} CHANGER_OBTAIN_SENSE, *PCHANGER_OBTAIN_SENSE;

```

An example of the **IOCTL\_CHANGER\_OBTAIN\_SENSE** command is

```

DWORD cb;
CHANGER_OBTAIN_SENSE sense_data;
DeviceIoControl(hDevice,
IOCTL_CHANGER_OBTAIN_SENSE,
NULL,
0,
&sense_data,
(long)sizeof(CHANGER_OBTAIN_SENSE),
&cb,
(LPOVERLAPPED) NULL);

```

## IOCTL\_MODE\_SENSE

This command is used to get the Mode Sense Page/Subpage.

```

/***** GENERIC SCSI IOCTLS *****/
#define IOCTL_IBM_BASE (('IBM' << 8) | FILE_DEVICE SCSI)

#define DEFINE_IBM_IOCTL(x) CTL_CODE(IOCTL_IBM_BASE, x, METHOD_BUFFERED, \
 FILE_READ_ACCESS | FILE_WRITE_ACCESS)
#define IOCTL_MODE_SENSE DEFINE_IBM_IOCTL(0x003)

```

The structure that is used for this IOCTL is

```
typedef struct _MODE_SENSE_PARAMETERS
{
 UCHAR page_code; /* [IN] mode sense page code */
 UCHAR subpage_code; /* [IN] mode sense subpage code */
 UCHAR reserved[6];
 UCHAR cmd_code; /* [OUT] SCSI Command Code: this field is set with */
 /* SCSI command code which the */
 /* device responded. */
 /* x'5A' = Mode Sense (10) */
 /* x'1A' = Mode Sense (6) */
 CHAR data[MAX_MODESENSEPAGE]; /* [OUT] whole mode sense data include header,
 block descriptor and page */
} MODE_SENSE_PARAMETERS, *PMODE_SENSE_PARAMETERS;
```

An example of the **IOCTL\_MODE\_SENSE** command is

```
DWORD cb;
MODE_SENSE_PARAMETERS mode_sense;
...
DeviceIoControl(gp->ddHandle0,
IOCTL_MODE_SENSE,
&mode_sense,
sizeof(MODE_SENSE_PARAMETERS),
NULL,
0,
&cb,
(LPOVERLAPPED) NULL);
```

---

## Variable and fixed block read/write processing

In Windows 200x, tape APIs can be configured to manipulate tapes that use either fixed block size or variable block size.

If variable block size is wanted, the block size must be set to zero. The `SetTapeParameters` function must be called specifying the `SET_TAPE_MEDIA_INFORMATION` operation. The function requires the use of a `TAPE_SET_MEDIA_PARAMETERS` structure. The `BlockSize` member of the structure must be set to the wanted block size. Any block size other than 0 sets the media parameters to fixed block size. The size of the block is equal to the `BlockSize` member.

In fixed block mode, the size of all data buffers used for reading and writing must be a multiple of the block size. To determine the fixed block size, the `GetTapeParameters` function must be used. Specifying the `GET_TAPE_MEDIA_INFORMATION` operation yields a `TAPE_GET_MEDIA_PARAMETERS` structure. The `BlockSize` member of this structure reports the block size of the tape. The size of buffers that are used in read and write operations must be a multiple of the block size. This mode allows multiple blocks to be transferred in a single operation. In fixed block mode, transfer of odd block sizes (for example, 999 bytes) is not supported.

When reading or writing variable sized blocks, the operation cannot exceed the maximum transfer length of the Host Bus Adapter. This length is the length of each transfer page (typically 4 K) times the number of transfer pages (if set to 16, the maximum transfer length for variable sized transfers is 64 K). This number can be modified by changing the *scatter-gather* variable in the system registry, but this action is not recommended because it uses up scarce system resources.

Reading a tape that contains variable sized blocks can be accomplished even without knowing what size the blocks are. If a buffer is large enough to read the data in a block, then the data is read without any errors. If the buffer is larger than a block, then only data in a single block is read and the tape is advanced to the next block.

The size of the block is returned by the read operation in the `*pBytesRead` parameter. If a data buffer is too small to contain all of the data in a block, then a couple of things occur. First, the data buffer contains data from the tape, but the read operation fails and `GetLastError` returns `ERROR_MORE_DATA`. This error value indicates that more data is in the block to be read. Second, the tape is advanced to the next block. To reread the previous block, the tape must be repositioned to the wanted block and a larger buffer must be specified. It is best to specify as large a buffer as possible so that this issue does not occur.

If a tape contains fixed size blocks, but the tape media parameters are set to variable block size, then no assumptions are made regarding the size of the blocks on the tape. Each read operation behaves as described. The sizes of the blocks on the tape are treated as variable, but happen to be the same size. If a tape has variable size blocks, but the tape media parameters are set to fixed block size, then the size of all blocks on the tape are expected to be the same fixed size. Reading a block of a tape in this situation fails and `GetLastError` returns `ERROR_INVALID_BLOCK_LENGTH`. The only exception is if the block size in the media parameters is the same as the size of the variable block and the size of the read buffer happens to be a multiple of the size of the variable block.

If `ReadFile` encounters a tapemark, the data up to the tapemark is read and the function fails. (The `GetLastError` function returns an error code that indicates that a tapemark was encountered.) The tape is positioned past the tapemark, and an application can call `ReadFile` again to continue reading.

---

## Event log

The Magstar® or `ibmtpxxx`, `ibmcgxxx`, and `Magchgr` device drivers log certain data to the Event Log when exceptions are encountered.

To interpret this event data, the user must be familiar with the following components:

- Microsoft Event Viewer
- The SDK and DDK components from the Microsoft Development Network (MSDN)
- Magstar and Magstar MP hardware terminology
- SCSI terminology

Several bytes of "Event Detail" data are logged under Source = Magstar or Magchgr (for Windows NT), or under Source = `ibmtpxxx` or `ibmcgxxx` (for Windows 2000; Windows Server 2003, Windows Server 2008, and Windows Server 2012).

The following description texts are expected:

- The description for Event ID (0) in Source (MagStar or `ibmtpxxx`) was not found. It contains the following insertion strings: `\Device\Tapex`.
- The description for Event ID (x) in Source (MagChgr) was not found.

The user must view the event data in Word format to properly decode the data.

[Table 1](#) and [Table 2](#) indicate the hexadecimal offsets, names, and definitions for Magstar or ibmtpxxx and ibmcgxxx event data. Magchgr event data has a unique format that appears later in this chapter.

Table 1. Magstar ibmtpxxx, and ibmcgxxx event data

| Offset    | Name                         | Definition                                                                                                                                                                                                                                                                                                                        |
|-----------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x00-0x01 | DumpDataSize                 | Indicates the size in bytes required for any DumpData the driver places in the packet.                                                                                                                                                                                                                                            |
| 0x02      | RetryCount                   | Indicates how many times the driver tried the operation and encountered this error.                                                                                                                                                                                                                                               |
| 0x03      | MajorFunctionCode            | Indicates the IRP_MJ_XXX from the driver's I/O stack location in the current IRP (from NTDDK.H).                                                                                                                                                                                                                                  |
| 0x0C-0x0F | ErrorCode                    | For the Magstar device driver, it is 0. For the Magchgr device driver, it is always 0xC00400B (IO_ERR_CONTROLLER_ERROR, from NTIOLOGC.H).                                                                                                                                                                                         |
| 0x10-0x13 | UniqueErrorValue             | Reserved                                                                                                                                                                                                                                                                                                                          |
| 0x14-0x17 | FinalStatus                  | Indicates the value that is set in the I/O status block of the IRP when it was completed or the STATUS_XXX returned by a support routine the driver called (from NTSTATUS.H).                                                                                                                                                     |
| 0x1C-0x1F | IoControlCode                | For the Magstar device driver, it indicates the I/O control code from the driver's I/O stack location in the current IRP if the MajorFunctionCode is IRP_MJ_DEVICE_CONTROL. Otherwise, this value is 0. For the Magchgr device driver, it indicates the I/O control code from the driver's I/O stack location in the current IRP. |
| 0x28      | Beginning of Dump Data       | The following items are variable in length. See the DDK and SCSI documentation for details.                                                                                                                                                                                                                                       |
| 0x38      | Beginning of SRB structure   | The SCSI Request Block (from NTDDK.H).                                                                                                                                                                                                                                                                                            |
| 0x68      | Beginning of CDB structure   | The Command Descriptor Block (from SCSI.H).                                                                                                                                                                                                                                                                                       |
| 0x78      | Beginning of SCSI Sense Data | (from SCSI.H). If the first word in this field is 0x00DF0000 (SCSI error marker) or 0x00EF0000 (Non-SCSI error marker), no valid sense information was available for this error.                                                                                                                                                  |

For example, ibmcgxxx logs the following error when a move medium is attempted and the destination element is full. Explanations of selected fields follow.

```
0000: 006c000f 00c40001 00000000 c004000b
0010: bcde7f48 c0000284 00000000 00000000
0020: 00000000 00000000 00000000 000052f4
0030: 00000000 00000000 004000c4 02000003
0040: 600c00ff 00000028 00000000 00000258
0050: 00000000 814dac28 00000000 bcde7f48
0060: 81841000 00000000 a5600000 00200010
0070: 00000000 00000000 70000500 00000058
0080: 00000000 3b0dff02 00790000 0000093e
0090: 00000000
```

Table 2. Magstar ibmtpxxx, and ibmcgxxx event data

| Field             | Value         | Definition                                                                                                                                                                                                                                    |
|-------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DumpDataSize      | 0x006C        | 6C hex (108 dec) bytes of dump data, beginning at byte 28 hex.                                                                                                                                                                                |
| RetryCount        | 0x00          | The first time that the operation is attempted (no retries).                                                                                                                                                                                  |
| MajorFunctionCode | 0x0F          | IRP_MJ_INTERNAL_DEVICE_CONTROL                                                                                                                                                                                                                |
| FinalStatus       | 0xC0000284    | STATUS_DESTINATION_ELEMENT_FULL                                                                                                                                                                                                               |
| IoControlCode     | 0x00000000    | -                                                                                                                                                                                                                                             |
| SRB               | 0x004000C4... | From NTDDK.H, the first word of the SRB indicates the length of the SRB (40 hex bytes, 64 dec bytes), the function code (0x00), and the SrbStatus (from SRB.H, 0xC4 = SRB_STATUS_AUTOSENSE_VALID, SRB_STATUS_QUEUE_FROZEN, SRB_STATUS_ERROR). |
| CDB               | 0xA5...       | From SCSI.H, the first byte of the CDB is the operation code. 0xA5 = SCSIOP_MOVE_MEDIUM.                                                                                                                                                      |
| Sense Data        | 0x70000500... | From SCSI.H, the first word of the sense data indicates the error code (0x70), the segment number (0x00), and the sense key (0x05, corresponding to an illegal SCSI request).                                                                 |

[Table 3](#) and [Table 4](#) contain definitions for event data that is logged under Magchgr.

Table 3. Magchgr event data

| Offset    | Name              | Definition                                                                                                                                                                                                                                                                                                                      |
|-----------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x00-0x01 | DumpDataSize      | Indicates the size in bytes required for any DumpData the driver places in the packet.                                                                                                                                                                                                                                          |
| 0x02      | RetryCount        | Indicates how many times the driver tried the operation and encountered this error.                                                                                                                                                                                                                                             |
| 0x03      | MajorFunctionCode | Indicates the IRP_MJ_XXX from the driver's I/O stack location in the current IRP (from NTDDK.H).                                                                                                                                                                                                                                |
| 0x0C-0x0F | ErrorCode         | For the Magstar device driver, it is 0. For the Magchgr device driver, it is always 0xC00400B (IO_ERR_CONTROLLER_ERR) (from NTIOLOGC.H).                                                                                                                                                                                        |
| 0x10-0x13 | UniqueErrorValue  | Reserved                                                                                                                                                                                                                                                                                                                        |
| 0x14-0x17 | FinalStatus       | Indicates the value that is set in the I/O status block of the IRP when it was completed or the STATUS_XXX returned by a support routine the driver called (from NTSTATUS.H).                                                                                                                                                   |
| 0x1C-0x1F | IoControlCode     | For the Magstar device driver, it indicates the I/O control code from the driver I/O stack location in the current IRP if the MajorFunctionCode is IRP_MJ_DEVICE_CONTROL. Otherwise, this value is 0. For the Magchgr device driver, it indicates the I/O control code from the driver's I/O stack location in the current IRP. |
| 0x29      | PathId            | SCSI Path ID                                                                                                                                                                                                                                                                                                                    |
| 0x2A      | TargetId          | SCSI Target ID                                                                                                                                                                                                                                                                                                                  |
| 0x2B      | LUN               | SCSI Logical Unit Number                                                                                                                                                                                                                                                                                                        |
| 0x2D      | CDB[0]            | Command Operation Code                                                                                                                                                                                                                                                                                                          |
| 0x2E      | SRB_STATUS        | See MINITAPE.H or SRB.H.                                                                                                                                                                                                                                                                                                        |
| 0x2F      | SCSI_STATUS       | See SCSI.H or a SCSI specification.                                                                                                                                                                                                                                                                                             |



| Offset    | Name                              | Definition                                                                                                                                                                                                                                  |
|-----------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x30-0x33 | Timeout Value                     | For the Magstar device driver, this value is always 0. For the Magchgr device driver, this value is the command timeout value in seconds.                                                                                                   |
| 0x38      | FRU or Sense Byte 14              | For the Magstar device driver, this value is the Field Replaceable Unit Code. For the Magchgr device driver, this value is Sense Byte 14.                                                                                                   |
| 0x39      | SenseKeySpecific[0]               | Indicates Sense Key Specific byte (Sense Byte 15).                                                                                                                                                                                          |
| 0x3A      | SenseKeySpecific[1] or CDB length | If valid sense data was returned, SenseKeySpecific[1] (Sense Byte 16) is displayed. Otherwise, the CDB length is displayed. See offset 0x3D to determine whether valid sense data is returned.                                              |
| 0x3B      | SenseKeySpecific[2] or CDB[0]     | If valid sense data was returned, SenseKeySpecific[2] (Sense Byte 17) is displayed. Otherwise, the CDB operation code is displayed. See offset 0x3D to determine whether valid sense data is returned.                                      |
| 0x3C      | Sense Byte 0                      | Indicates the first byte of returned sense data.                                                                                                                                                                                            |
| 0x3D      | Sense Byte 2                      | Indicates the second byte of returned sense data. This byte contains the Sense Key and other flags. If the byte is set to 0xDF (SCSI Error Marker) or 0xEF (Non-SCSI Error Marker), no valid sense information was available for the error. |
| 0x3E      | ASC or SRB_STATUS                 | Indicates Sense Byte 12, if there was valid sense information. Otherwise, the SRB status value is given here. See offset 0x3D to determine whether valid sense data is returned.                                                            |
| 0x3F      | ASCQ or SCSI_STATUS               | Indicates Sense Byte 13, if there was valid sense information. Otherwise, the SCSI status value is given here. See offset 0x3D to determine whether valid sense data is returned.                                                           |

For example,

```
0000: 0018000f 006c0001 00000000 00000000
0010: 00000000 c0000185 00000000 00000000
0020: 00000000 00000000 00000300 0015c402
0030: 00000000 00000000 f50ac607 700b4b00
```

Table 4. Magchgr event data

| Field                                  | Value      | Definition                                                            |
|----------------------------------------|------------|-----------------------------------------------------------------------|
| DumpDataSize                           | 0x0018     | -                                                                     |
| RetryCount                             | 0x00       | -                                                                     |
| MajorFunctionCode                      | 0x0F       | IRP_MJ_INTERNAL_DEVICE_CONTROL                                        |
| FinalStatus                            | 0xC0000185 | STATUS_IO_DEVICE_ERROR                                                |
| IoControlCode                          | 0x00000000 | -                                                                     |
| PathId                                 | 0x00       | -                                                                     |
| TargetId                               | 0x03       | -                                                                     |
| LUN                                    | 0x00       | -                                                                     |
| CDB[0]                                 | 0x15       | Mode Select, Byte 6                                                   |
| SRB_STATUS                             | 0xC4       | SRB_STATUS_AUTOSENSE_VALID, SRB_STATUS_QUEUE_FROZEN, SRB_STATUS_ERROR |
| SCSI_STATUS                            | 0x02       | Check condition                                                       |
| FRU                                    | 0xF5       | -                                                                     |
| Sense Key Specific Sense Bytes 15 - 17 | 0x0AC607   | -                                                                     |
| Sense Byte 0                           | 0x70       | -                                                                     |
| Sense Key Sense Byte 2                 | 0xb4       | -                                                                     |
| ASC                                    | 0x4B       | -                                                                     |
| ACSQ                                   | 0x00       | -                                                                     |

## 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 may send license inquiries, in writing, to:

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

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

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 cannot apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes are incorporated in new editions of the publication. IBM may make improvements and/or changes in the products and/or programs 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.

The ITDT-SE and ITDT-GE software uses Henry Spencer's regular expression library that is subject to the following copyright notice:

"Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved. This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice cannot be removed or altered.

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries (or regions), or both:

|              |               |                 |                       |
|--------------|---------------|-----------------|-----------------------|
| AIX®         | IBMLink       | RS/6000®        | System z®             |
| AIX 5L       | Magstar®      | S/390®          | Tivoli®               |
| FICON®       | Micro Channel | StorageSmart    | TotalStorage™         |
| HyperFactor® | Netfinity     | System i®       | Virtualization Engine |
| i5/OS        | POWER5        | System p        | xSeries               |
| iSeries      | ProtecTIER®   | System Storage® | z9                    |
| IBM          | pSeries       | System x        | zSeries               |

Adobe and Acrobat are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Itanium, and Pentium are trademarks of Intel Corporation in the United States, other countries (or regions), or both.

Java™ and all Java-based trademarks are trademarks of Oracle, Inc. in the United States, other countries, or both.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and Windows 2000 are 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).

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

## How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

To submit any comments about this publication or any other IBM storage product documentation:

- Send your comments by email to [ibmkc@us.ibm.com](mailto:ibmkc@us.ibm.com). Be sure to include the following information:
  - Exact publication title and version
  - Page, table, or illustration numbers that you are commenting on
  - A detailed description of any information that should be changed

## Terms and conditions for knowledge centers

Permissions for the use of these publications are granted subject to the following terms and conditions.

### IBM Privacy Policy

We intend to protect your personal information and to maintain its integrity. IBM® implements reasonable physical, administrative, and technical safeguards to help us protect your personal information from unauthorized access, use, and disclosure. For example, the products only provide IBM data about the asset usage and configuration and do not reflect private use of the asset. When diagnostics are required to be sent to IBM, and a problem is submitted, that data is routed directly to a secured infrastructure. Only individuals with a need to know are given access while working to resolve your problem. When appropriate, we also require that our suppliers protect such information from unauthorized access, use, and disclosure.

Visit the IBM Privacy Policy for additional information on this topic at <https://www.ibm.com/privacy/details/us/en/>.

## Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

## Personal use

---

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You cannot distribute, display, or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

## Commercial use

---

You can reproduce, distribute, and display these publications solely within your enterprise provided that all proprietary notices are preserved. You cannot make derivative works of these publications, or reproduce, distribute, or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## Rights

---

Except as expressly granted in this permission, no other permissions, licenses, or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions that are granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You cannot download, export, or reexport this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM trademarks

---

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at <http://www.ibm.com/legal/copytrade.shtml>.