



Advanced Path Failover Driver Installation Guide

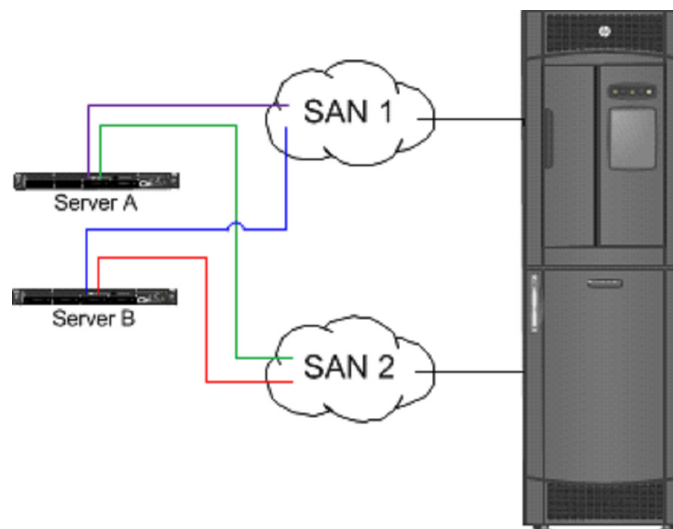
Overview

Introduction to Advanced Path Failover

Advanced Path Failover is a feature that uses capabilities in the HP LTO-6 tape drives and the libraries in which they are installed, combined with kernel mode drivers running on a host system to provide path failover when multiple paths are available to a tape drive or to a library controller.

An example configuration multi-path configuration is shown in [Figure 1](#).

Figure 1 Example Multi-path
Configuration



In this example configuration two different servers -designated as Server A and Server B - each have two different host interface ports connected to two different SANs. Each SAN is connected to the tape library. The connection between the library and the SAN in this example is a bundle of connections



which contains a connection to one port on each drive in the library. All of the drives in the library have two ports with one port connected into SAN 1 and the second port connected into SAN 2. The library in this example has two different drives which are both configured to provide a library control path. Each drive that is configured to provide a library control path will connect to the SAN as two devices, a tape drive and a library controller, at two different SCSI logical units.

The list of paths that are available at one of the servers is shown in Table 1. In this table the SAN that contains the path is listed in the first column and the following columns show what device is being addressed, which port on the drive is being accessed, an example SCSI address, and the worldwide identifier of the addressed logical unit. The unique portion of the worldwide identifier has been highlighted in [Table 1](#).

Table 1 Example hardware path list

SAN	Addressed Logical Unit	Port	Example SCSI Address	Example Logical Unit Worldwide Identifier
1	Tape drive 1	Port A	ID 1 LUN 0	50:01:10:a0:00:00:00:01
1	Library Controller	Port A	ID 1 LUN 1	50:01:10:a0:00:00:00:02
2	Tape drive 1	Port B	ID 2 LUN 0	50:01:10:a0:00:00:00:01
2	Library Controller	Port B	ID 2 LUN 1	50:01:10:a0:00:00:00:02
1	Tape drive 2	Port A	ID 3 LUN 0	50:01:10:a0:00:00:00:03
1	Library Controller	Port A	ID 3 LUN 1	50:01:10:a0:00:00:00:02
2	Tape drive 2	Port B	ID 4 LUN 0	50:01:10:a0:00:00:00:03
2	Library Controller	Port B	ID 4 LUN 1	50:01:10:a0:00:00:00:02

In this example the server is able to evaluate the worldwide identifiers to determine that there are two paths to each of two different tape drives and four paths to one library controller.

In a typical multi-path configuration all of the paths for each device are available to the application which must resolve redundant paths and choose a preferred path.

When advanced path failover drivers are installed on the server, the drivers query each device for support of advanced path failover. If the device supports advanced path failover, then the drivers will select a preferred path for each device and establish a connection using the preferred path. Only one path will be visible to the application.

Advanced Path Failover Overview

The high level operation of advanced path failover is the same for both the data path (paths to a tape drive) and for the control path (paths to a library

controller). If the failover drivers for a library control path is able to switch to an alternate path that addresses the same control path drive, then failover for a library control path is identical to failover for the data path. Library control path failover drivers perform a few extra steps during failover when the path must be changed to a path that addresses the library control path using a different drive.

Driver Operation During Device Open

When an application requests that a connection to a device be opened, the failover driver will connect to the selected device, and if that device supports advanced path failover, will negotiate for a connection (a nexus using T10 standards terminology) to that device. Computers running Windows will open a connection to each device when the first command is received from a client application. Computers running HP-UX and Linux don't open a connection to the device until an application opens the device for reading and/or writing.

Each device supports up to 32 simultaneous failover connections. If a failover connection is available, a connection between the host driver and the device is created. If no failover connection is available, then the connection that has been idle for the longest is closed and a new connection is created. The server connected to the idle session that was terminated is notified that the failover connection has been terminated. The advanced path failover driver on that system automatically creates a new connection the next time a command is sent to the device.

Environments using advanced path failover should be designed with 32 or fewer active hosts per device for optimal performance. Host should be configured so that they do not send unnecessary polling commands to the device.

Driver Operation During Normal Communications

The advanced path failover drivers pass commands through without any command translation and with minimal additional processing in normal operation to retain the maximum possible performance. The advanced path failover drivers do not introduce additional SCSI commands such as commands for reading position which would introduce delays.

Path Failure Detection

The advanced path failover drivers use notifications from the SCSI subsystem that report link failures immediately following a path failure allowing recovery to happen as quickly as possible. This results in most recoveries completing before the standard command timeout rather than waiting for a command timeout. In some operating systems the path failure notification is received immediately after the failure and the failover drivers are able to perform path failure recovery even if there are no outstanding commands. In other operating systems the advanced path failover drivers are only notified of a path failure when a command is transmitted over that path.

Path Failure Recovery

Following detection of a path failure, the advanced path failover drivers query a path verification feature in the LTO tape drive to test paths until a valid path to the device is detected. The path verification feature allows rapid detection of

failed and valid paths without timeouts or hardware specific notifications. After a new path has been identified, the advanced path failover drivers send a command to the drive using the new path to notify the device that a path has failed, indicate which connection failed, and to provide state information.

Upon notification that the path has been changed, the drive automatically transfers all available settings and information from the failed connection to the new connection. It uses the state information provided to synchronize the target state with the device driver state. The drive then notifies the device driver that it has successfully synchronized state, including making any physical position changes necessary to position the tape to the correct logical position.

After receiving notification that the state is synchronized between the advanced path failover drivers and the device, the advanced path failover drivers are able to take the steps necessary to recover any commands that were outstanding at the time of the failure. For most commands recovery is accomplished by resending the original command.

Notifying the target device of the path change and performing the state synchronization in the target device removes complex state recovery algorithms from the driver and removes the risk of incorrect tape positioning during state recovery, resulting in a higher performance, lower complexity, and less risky path failover method than a traditional driver where all recovery is performed by the driver.

Active and Passive Control Path Drives

In an environment configured for advanced path failover at least two tape drives should be configured to provide a path to the library controller. Some library state information is retained in the drive hosting the library controller so each server that requests a connection to the library controller is required to create a connection using a path through the same hosting drive. The library management interface provides a method for the administrator to select the preferred control path drive (i.e., the active control path drive), and the advanced path failover drivers will query each available control path drive to determine which is the preferred control path drive before requesting a connection.

The tape drive configured as an active control path drive accepts commands, such as MOVE MEDIUM for the changer device. Another drive is configured as a passive control path drive. This means that the SMC devices presented by this tape drive accept changer device discovery commands, such as INQUIRY, but rejects commands, such as MOVE MEDIUM. The advanced path failover drivers automatically selects an active path to the medium changer and automatically reconfigures which drive is the active control path drive when reconfiguration is necessary during failover. The tape library user interface shows which control path drive is the current active control path drive.

Library Controller Path Failure Recovery

In most cases the library controller path failure recovery is the same as the tape drive path failure recovery. If all paths to the drive hosting the library controller have failed, then the advanced path failover drivers are capable of opening a connection to the library controller through another tape drive.

The library control path is hosted by a tape drive, which maintains some state information on behalf of the library controller. The state information maintained by the tape drive includes reservations, medium removal restrictions, and may contain special mode settings. Any time the library state changes, the library notifies the advanced path failover drivers that a state change has occurred and then the advanced path failover drivers retrieve a cache of the state information.

When all paths to the drive that was hosting the library control path fail, the advanced path failover drivers will connect to a different drive and configure the new drive as the preferred control path drive. After activating a passive control path drive the path failover drivers will download the state information from the previous control path drive. The new control path drive notifies the library controller that the preferred control path drive has changed and the library disables the previous control path drive. If any servers are still connected to the original control path drive, they are notified that the preferred control path drive has changed and the advanced path failover drivers on those servers automatically change the preferred path for future commands to the new preferred control path drive.

Windows Driver Theory of Operation

Overview

The advanced path failover drivers support Data Path Failover (DPFO) and Control Path Failover (CPFO) on HP LTO-6 drives integrated into Quantum automation.

A failover device driver must identify devices accessible over the different paths available and associate paths with devices so that it is able to mask duplicate paths and present a single path to an ISV application.

As devices are discovered by the operating system and the device objects are passed into the driver for initialization, the driver determines if each device is a tape or changer device and is a candidate for failover. If the device is a candidate for failover, the driver determines the serial number (SN) and worldwide node name (WWNN) of the device. If the device is capable of advanced path failover, the driver determines whether failover is enabled for the device. If the device does support failover and failover is enabled, then the driver will mask all duplicate paths and present a single path to application clients. If the device does not support failover, failover is not enabled, or an error occurred while attempting to determine the failover-enabled state, then the driver will not modify the presentation of paths to the device to applications.

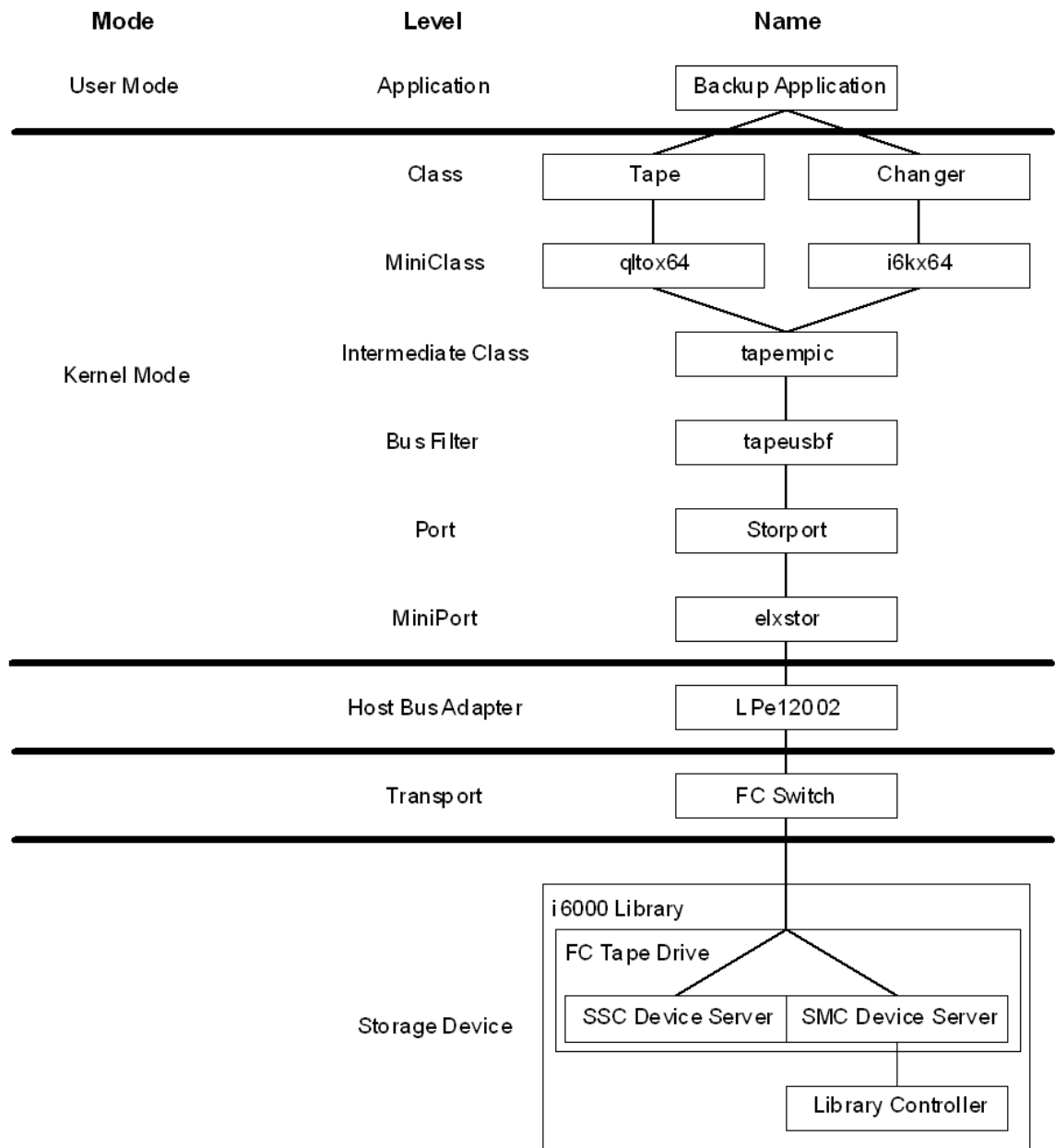
Driver Components

Windows advanced path failover drivers consists of two drivers: a storage bus filter driver and an intermediate class driver. The storage bus filter driver processes OS device notifications (PnP notifications) to identify available paths to tape and media changer devices capable of supporting advanced path failover. The intermediate class driver then manages the active paths to those devices.

This architecture follows the architecture of Microsoft's MPIO. MPIO only supports disk devices and cannot be used directly.

Figure 2 shows the organization of the Windows drivers and the system components when an Emulex LPe12002 host bus adapter is installed. In this example, the tape drive presents both SSC (tape) and SMC (media changer) device servers. Not all drives in the library present an SMC device server.

Figure 2 Driver and System Organization



Storage Bus Filter Driver

The storage bus filter driver is named `tapeusbfilter`. It manages the OS notifications indicating that devices have been added and removed. The storage bus filter driver bus enumerator is usually `PCI`; for `iSCSI`, its bus enumerator is `ROOT`. It attaches as an upper filter to all bus drivers (i.e. HBA drivers). It then monitors the creation of raw devices, looking for supported failover capable devices. Requests to all other devices are passed through unchanged.

When a supported device is seen, the hardware ID of the device is changed to a value that the intermediate class driver will recognize, but that the rest of the system will not. This special hardware ID prevents the standard tape and changer class drivers from attaching to the raw paths.

Multi-path Intermediate Class Driver

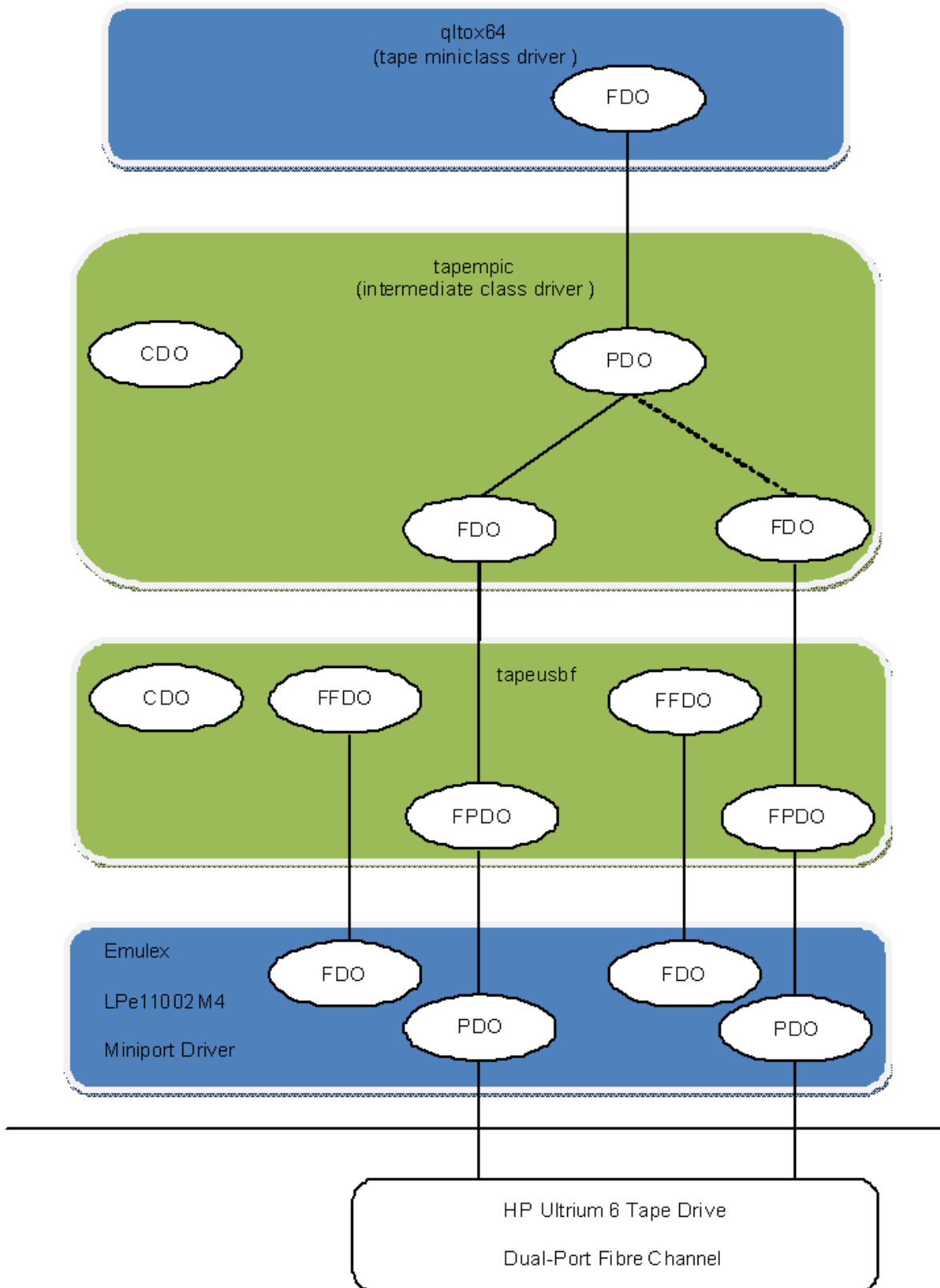
The intermediate class driver (`tapempic`) has two halves. The upper half is a virtual bus driver that manages the paths to all devices supporting failover. The lower half is a multi-path function driver for the custom PDOs created by the storage bus filter driver. The custom PDOs that the lower half function driver attaches to are actually the changer and tape FPDOs created by the storage bus filter driver. The upper half has a bus enumerator of `ROOT`. The lower half has a bus enumerator of `APFO` (advanced path failover).

The lower half of this driver acts like a class driver, in that it attaches to the devices presented by the storage bus filter driver. The upper half acts like a virtual bus driver, presenting abstractions of devices to the layers above. For example, a single tape device may have many paths to it, but only a single virtual tape device is presented to the layers above, and the driver transparently manages the multiple paths to the physical tape device.

[Figure 3](#) on page 8 shows the various device objects presented by the drivers for a single tape drive. The abbreviations are:

- **PDO**: Physical device object
- **FDO**: Functional device object
- **FPDO**: PDO created by bus filter driver
- **FFDO**: FDO created by bus filter driver
- **CDO**: Control device object; used by diagnostic interface

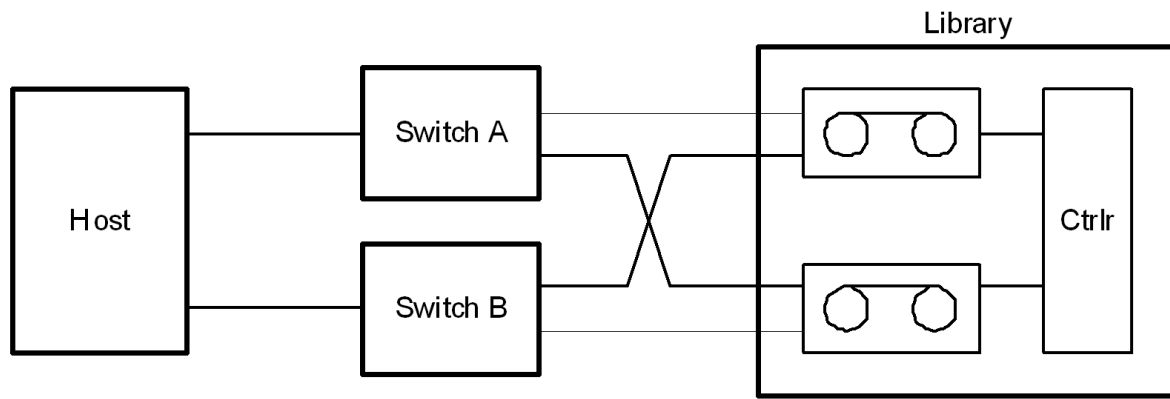
Figure 3 Drivers and device objects



Device Manager View

In the system shown in [Figure 4](#) on page 9, the library has two tape drives, each with two primary ports attached to a different switch. Each drive contains an SSC device (tape drive) and an SMC device (media changer). While each drive is connected to the library controller, only one of the drives will have its SMC device configured to send commands to the library at any time.

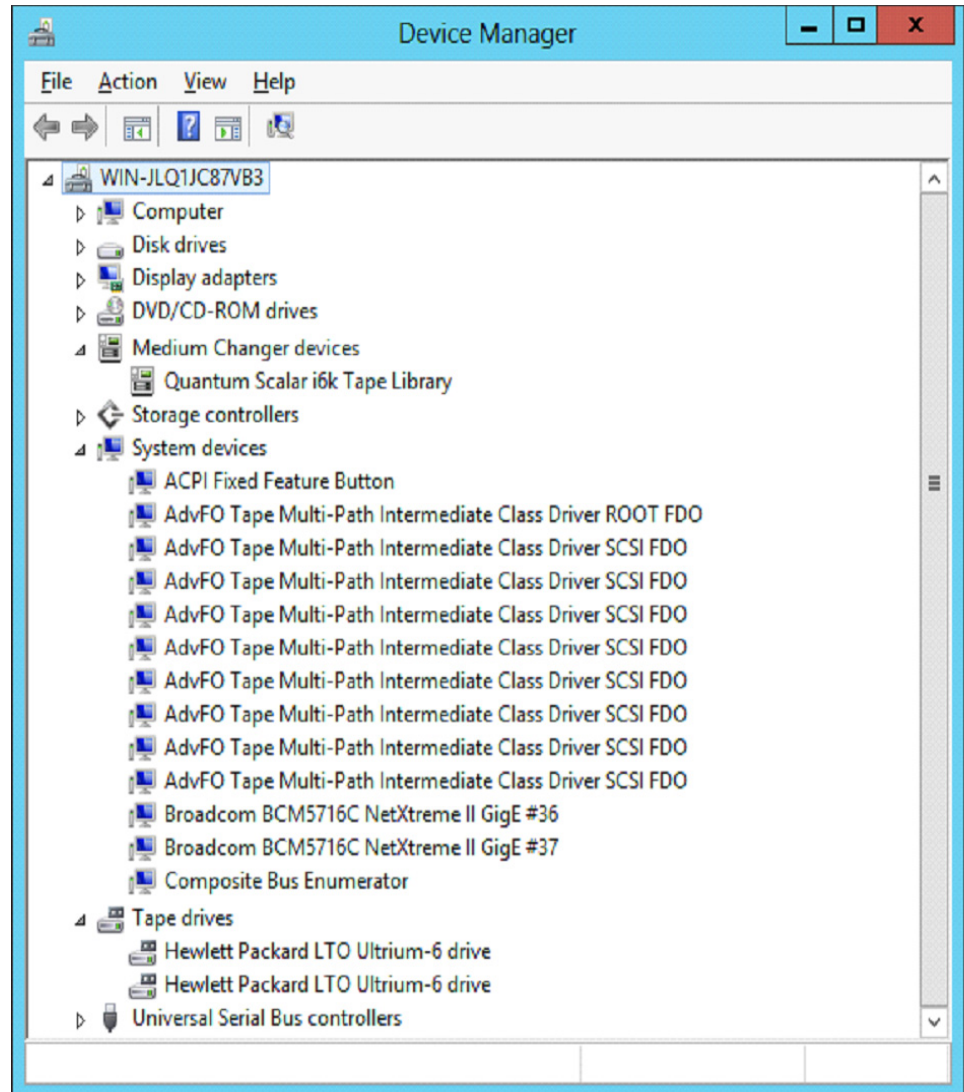
Figure 4 Example Library Configuration



Through switch A, the failover drivers have available one path to each of the two tape drives. They have available another path to each drive through switch B. Because each of those four paths can potentially connect to an SSC device and an SMC device, at boot time the drivers discover a total of eight potential paths to devices. This results in the Windows Device Manager view that is shown in [Figure 5](#) on page 10. The failover paths appear under **System devices**. A virtual bus named **AdvFO Tape Multi-Path Intermediate Class Driver ROOT FDO** is created to enumerate the multi-path capable devices. Each of the eight paths appears as **AdvFO Tape Multi-Path Intermediate Class Driver SCSI FDO**.

The drivers make one path active to the SSC device in each drive, and one path active to the SMC device in only one of the drives. This is why each physical device appears once in the Device Manager. For example, under **Tape drives** there are two instances of **Hewlett Packard LTO Ultrium 6 drive** and under **Medium Changer** devices there is one instance of **Quantum Scalar i6k Tape Library**.

Figure 5 Device Manager window



Device Firmware

These two Windows drivers work in conjunction with special firmware running in the changer and tape devices. The special firmware ensures that the host computer and devices stay synchronized on state and position information. The advanced path failover drivers hide the details of this from the rest of the system. The objective is for the rest of the system, including all applications running on the system and all other device drivers, to see a single changer or tape device but not the redundant paths to the failover-enabled devices. If a path failure occurs it should be transparent to the rest of the system including applications. If a path fails, the drivers work with the device to transfer communication to a new path and recover the command that was in process when the path failed.

Additional Sense Codes

Tape drives supporting advanced path failover use vendor specific SCSI additional sense codes (ASCs) to report certain conditions to the failover drivers.

These additional sense codes are handled by the failover drivers in normal operation, and should not be visible to applications.

Because these ASCs may be visible in device logs and diagnostic tools, they are listed in [Table 2](#).

Table 2 Advanced failover vendor specific additional sense codes

ASC	ASCQ	Description
82h	93h	FAILOVER SESSION SEQUENCE ERROR
82h	94h	FAILOVER COMMAND SEQUENCE ERROR
82h	95h	DUPLICATE FAILOVER SESSION KEY
82h	96h	INVALID FAILOVER KEY
82h	97h	FAILOVER SESSION RELEASED
82h	98h	SMC STATE CHANGED
82h	99h	FAILOVER SMC DEVICE SERVER MOVED

Enable Advanced Path Failover in the Library

Tape libraries that support advanced path failover provide a mechanism in the library user interface for enabling advanced path failover by installing a license. See the library user guide for specific instructions.

Advanced Path Failover Drivers for Windows

Version Support

Advanced path failover drivers are available for the following versions of Windows:

Table 3 Release Driver Installers

Windows Version	Quantum Installer File
Windows 2012 R2 Windows 2012 Windows 2008 R2 Windows 2008 x64	apf.exe

Install Drivers

The procedure for installing the drivers is:

- 1 Begin installation by double-clicking on the installer, e.g. **apf.exe**. This first installs the tape upper bus filter driver.
- 2 Restart when requested.
- 3 After the system restarts, the installer continues with installation of the tape multi-path intermediate class driver.

Note: Installation requires Windows Installer version 4.5 or later. If it is not present, then the installer reports the errors:

- Error 0x80070643: Failed to install MSI package
- Error 0x80070643: Failed to execute MSI package
- Error 0x80070643: Failed to configure per-machine MSI package.

If these errors appear, then download the Windows Installer appropriate for your version of Windows using the following link:

<http://www.microsoft.com/en-us/download/details.aspx?id=8483>

After installing the Windows Installer, then return to step 1.

The installation process creates a directory:

C:\Program Files\Quantum\Advanced Failover Drivers

Uninstall Drivers

For Windows 2008 or 2012, uninstall the drivers using the **Programs** control panel. Select the **Advanced Failover Drivers** entry, click on the **Uninstall/Change** button, and follow the prompts. Previous versions of the drivers must be uninstalled in this manner when upgrading to newer drivers.

Debug Failover Operation

There are five means of obtaining diagnostic information about the operation of the advanced path failover mechanism:

- Examine the library tickets
- Examine the **Windows System** log for entries by the drivers
- Examine the device manager view of tape drives and media changers
- The **Advanced Failover Diagnostic** application can be used to display the state of failover operation of the drivers
- The **Advanced Failover Diagnostic** application can be used to enable logging. When logging is enabled, the drivers will write run-time operational information to files that can be sent to Quantum Corporation for analysis

Library Support Tickets

When the library detects a path failover, it will generate an appropriate support ticket. Tickets are summarized in [Table 4](#).

Table 4 Library Support Tickets

Event	Description
Library Control Path Failover	A library control path failed and a failover operation to a redundant control path drive succeeded.
Library Control Path Failover Failure	A library control path failed and a failover operation to a redundant control path did not succeed. The library control path is not operational.
Drive Data Path Failover	A drive data path connection failed and a failover operation to the redundant drive port succeeded.
Drive Data Path Failover Failure	A drive data path connection failed and a failover operation to the redundant data port failed. The drive data path is not operational.
Drive Control/ Communication Failure	A drive communication failure occurred which prevents drive control and library control path operations.

Windows System Log

The drivers create entries in the Windows system log when they are first started and whenever any exceptional conditions occur. These log entries are summarized in [Table 5](#). The symbols beginning with % are strings filled in by the operating system when the log entry is generated.

Table 5 Event Log Entries

Event	Severity	Message
Driver loaded	Informational	Path failover driver loaded
Driver unable to load	Error	Path failover driver unable to load
Path failure detected	Error	Path failure detected on SCSI Port %s Bus %3 Target Id %4 Logical Unit %5
Path removal reported by Windows	Error	Path removal reported on SCSI Port %1 SCSI Bus %2 Target Id %3 Logical Unit %4.
Device serial number for failing path	Informational	Device ID: %1
Device Names for failing path	Informational	Device WWNN: %1-%2 Device WWPNN: %3-%4 Host WWPNN: %5-%6.

Event	Severity	Message
Error reported by failing path	Informational	Device error: NT status = %1. SCSI Status = %2, Sense Key = %3, ASC/ASCQ = %4/%5, Drive Error Code = %6/%7.
Failover session established	Informational	Failover session established on SCSI Port %2 SCSI Bus %3 Target Id %4 Logical Unit %5
Failover session establishment error	Error	Failover session establishment error on SCSI Port %2 SCSI Bus %3 Target Id %4 Logical Unit %5. Reason code %6.
No alternate paths available	Error	No alternate paths for device ID: %1
Alternate path selected after a path failure	Informational	Alternate Path selected: SCSI Port %1 SCSI Bus %2 Target Id %3 Logical Unit %4.
Failure to transition a passive control path drive to active	Error	License key error: NT status = %1. SCSI Status = %2, Sense Key = %3, ASC/ASCQ = %4/%5, Drive Error Code = %6/%7.
Failure to notify device of a failover operation (NCN failure)	Error	Nexus change notification failure: NT status = %1. SCSI Status = %2, Sense Key = %3, ASC/ASCQ = %4/%5, Drive Error Code = %6/%7.
Hardware error from device on NCN	Informational	Nexus change notification will retry after: Sense Key = %1, ASC/ASCQ = %2/%3.
Unrecoverable error in failover sequence negotiation after NCN failure.	Informational	Recovery aborted, sense data modified: Sense Key = %1, ASC/ASCQ = %2/%3.
Sense information from a device requires a new failover session be established	Informational	A new session will be created after: NT status = %1. SCSI Status = %2, Sense Key = %3, ASC/ASCQ = %4/%5, Drive Error Code = %6/%7.
Unable to retrieve logged in host table from a tape drive	Informational	Retrieve logged in host name failed: NT status = %1. SCSI Status = %2, Sense Key = %3, ASC/ASCQ = %4/%5, Drive Error Code = %6/%7.

The reason code for a failover session establishment error is a Windows system error number. Because there is no particular action that the user can take that depends upon the error code, the values are not described here. The reason code should be included with the rest of the message in any report sent to Quantum.

The log is easier to use if irrelevant events are filtered out:

- 1 Open the **System** log.
- 2 In the **Actions** panel (on the right) select **Filter Current Log...**
- 3 In the **Event sources** drop-down menu, click the check boxes for:
 - tapeusb
 - tapempic, and
 - UserPnp
- 4 Click **OK** to dismiss the **Filter Current Log**. The log will now show only events relevant to device discovery and APFO failover operation.

Diagnostic Application

The **Advanced Failover Diagnostic** application communicates with the drivers to obtain the state of failover sessions. It can be used not only to debug problems leading to fail overs, but also to confirm correct installation of the failover drivers.

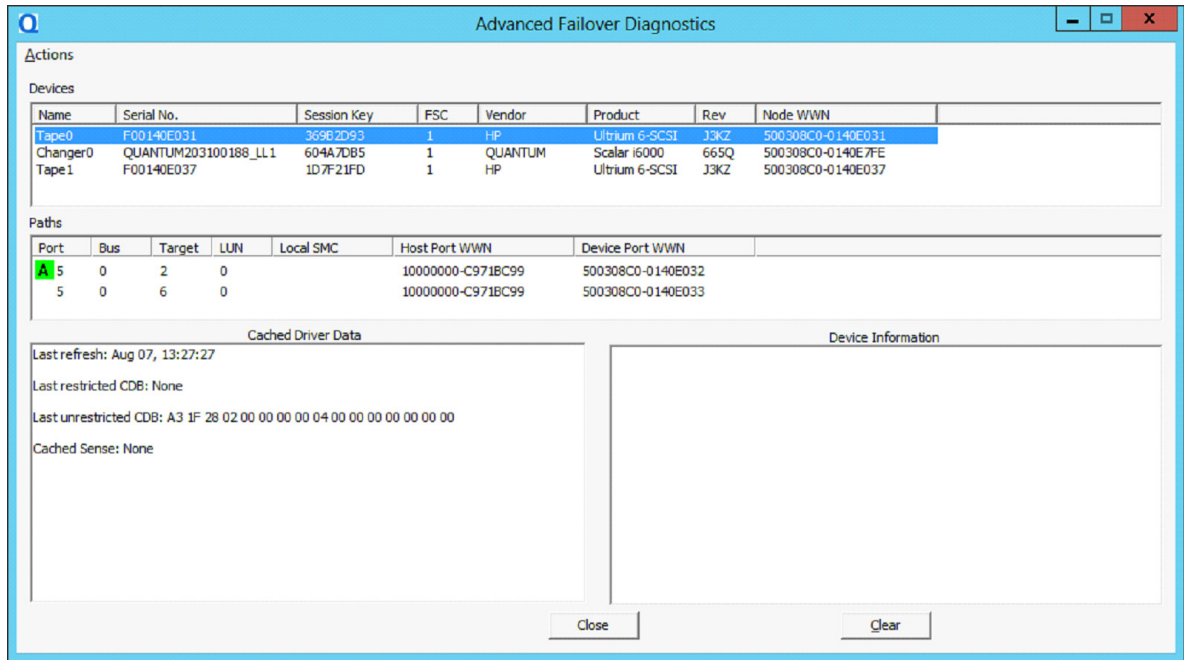
For Quantum installations, the application consists of two files, AdvFoDgn.exe and TapeMpic.dll. These are installed when the driver is installed. The files will be located in:

C:\Program Files (x86)\Quantum\Advanced Failover Drivers\AdvFoDgn

Operation

The diagnostic application is started by double-clicking its icon. It displays the window shown in [Figure 6](#) on page 16. The window contains four panes.

Figure 6 Advanced failover diagnostic application



The **Devices** pane lists the tape drives and media changers that are associated with the path failover drivers

The **Paths** pane shows information about the media changer device which is selected in the Devices pane. In [Figure 6](#), the information is for the media changer named **Changer0**. The capital **A** in the **Port** column indicates the active control path to the SMC device server that is currently selected, and the lower case **a** indicates an alternate active control path to the SMC device server. All other paths are passive, i.e., they pass through a passive control path drive (see [Active and Passive Control Path Drives](#) on page 4).

If the selected device is a tape drive, then the capital **A** indicates the active data path, and there will be no lower case **a**.

The **Cached Driver Data** pane shows the last state of the selected device which was reported by the failover drivers to the diagnostic application. Because changes in the drivers' information about a device are not automatically sent to the diagnostic application, it is necessary to perform a refresh (see [Refresh](#) on page 17) to get the latest state of the drivers.

The **Device Information** pane contains data requested from the tape and changer devices when various actions ([Figure 8](#) on page 18) are performed.

Actions

There are several actions which can be performed by the diagnostic application:

- Refresh failover drivers' state information held by the diagnostic application.
- Request current failover information from tape and changer devices.

- Clear reservations on changer devices which had been set by hosts that are now offline.

Refresh

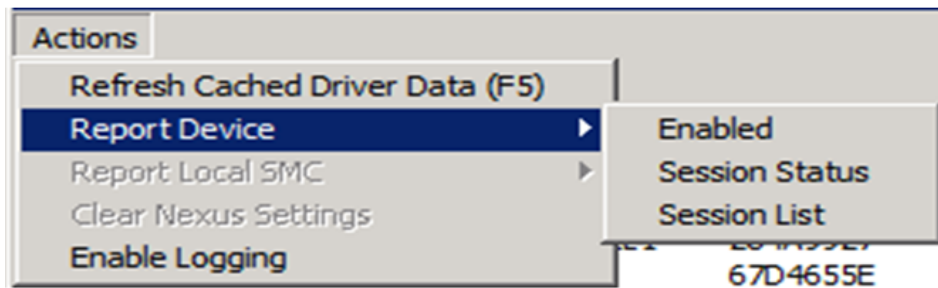
The information in the Cached Driver Data pane can be refreshed by selecting **Actions > Refresh Cached Driver Data (F5)**.

Device information

The **Actions > Report Device** menu has three sub-menu items (see [Figure 8](#) on page 18) which report information on the drive or media changer selected in the **Devices** pane. The data is displayed in the Device Information pane.

Note: Two actions pertaining only to changer devices are grayed out when a tape device is selected.

Figure 7 Report Device Menu Items



The **Device Information** pane in [Figure 8](#) on page 18 shows the results of performing the **Enabled**, **Session Status**, and **Session List** actions in sequence.

Figure 8 Report Device Information

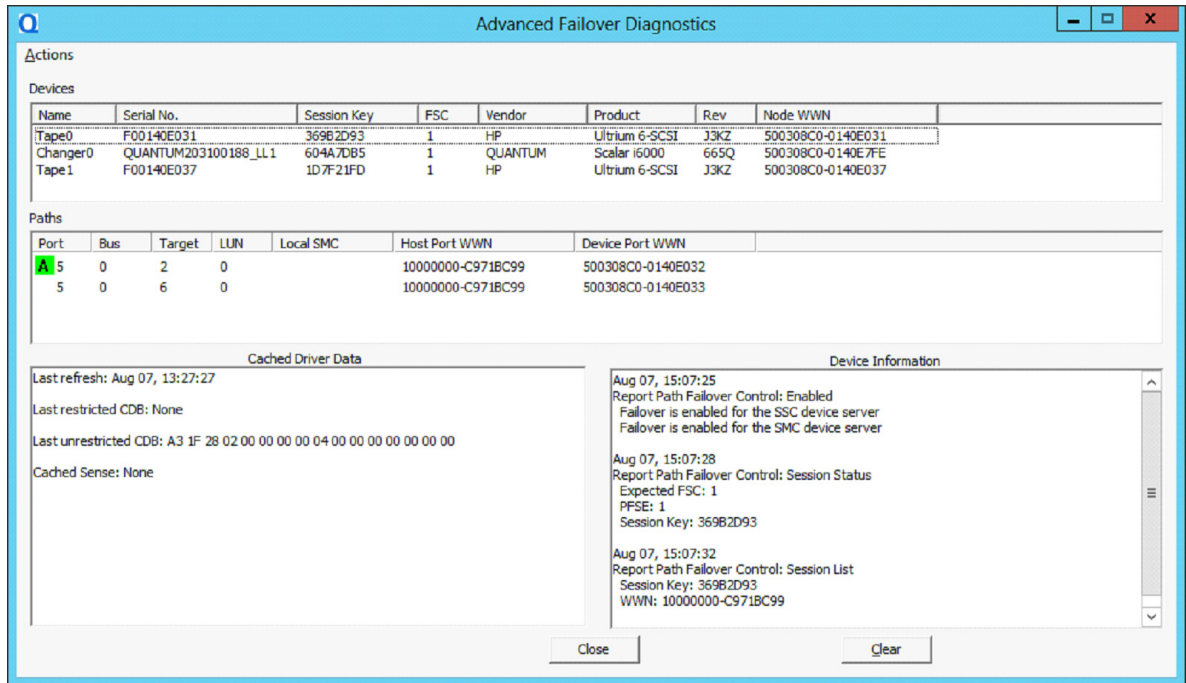


Table 6 shows the meanings of the values reported for the **Session Status** and **Session List** actions.

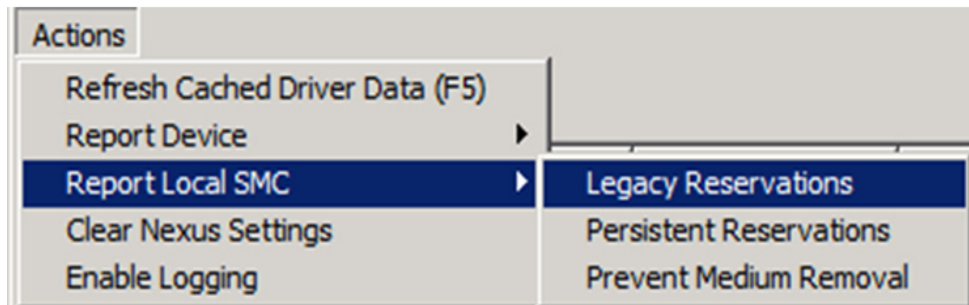
Table 6 Device information

Menu Item	Data
Session Status	<p>Expected FSC: A command indicator used for management of commands that require synchronization following a link failure.</p> <p>PFSE:</p> <p>1 - The device server is using failover via this device server. 0 - The device server is not using failover via this device server.</p> <p>Failover Session Key: The key for the path (I_T_L nexus) over which the driver is communicating with the device.</p>
Session List	<p>Each entry in the session list contains information about one failover session:</p> <ul style="list-style-type: none"> Failover Session Key Initiator Worldwide Name

Local SMC Device Server Information

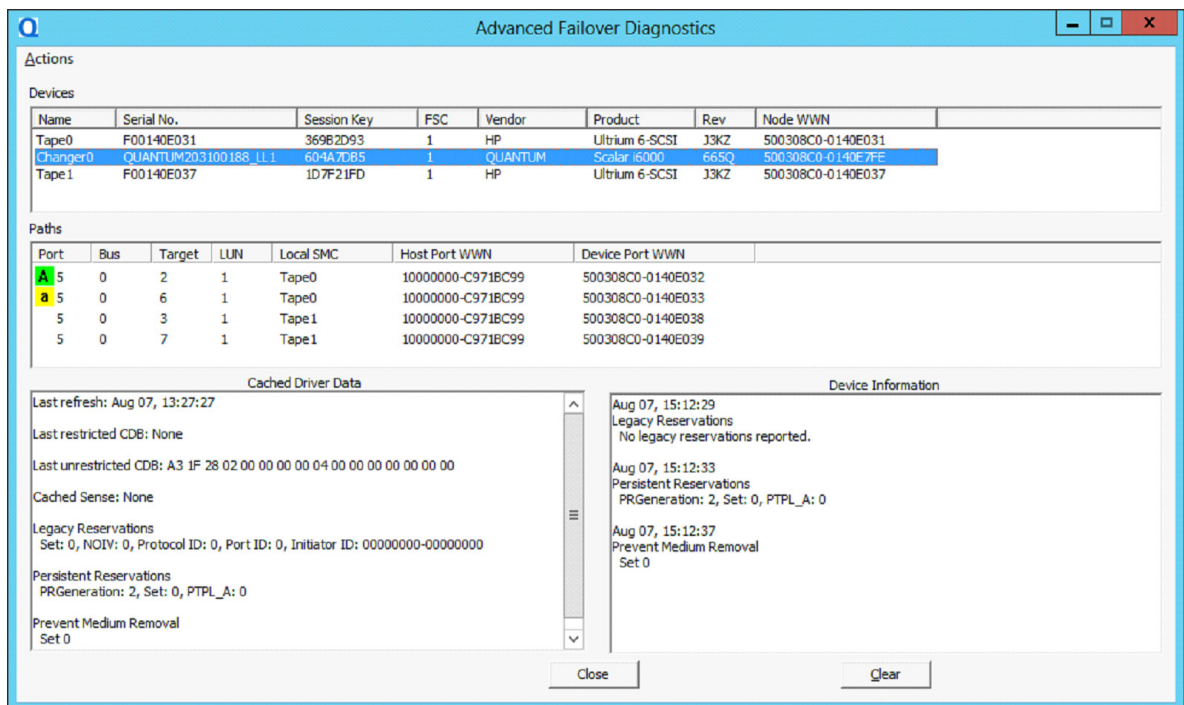
The **Actions > Report Local SMC** menu has three sub-menu items (see [Figure 9](#)) which report information on the media changer selected in the **Devices** pane. The data is displayed in the **Device Information** pane.

Figure 9 Report Local SMC menu items



The **Device Information** pane in [Figure 10](#) shows the results of performing the **Legacy Reservations**, **Persistent Reservations**, and **Prevent Medium Removal** actions in sequence.

Figure 10 Report Local SMC information



[Table 7](#) on page 20 shows the meanings of the values reported for the actions.

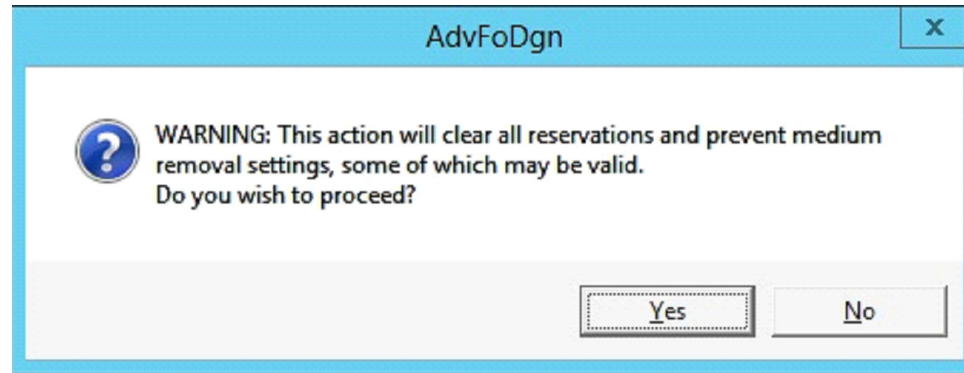
Table 7 Local SMC device server information

Menu Item	Data
Legacy Reservations	SET: 1: An initiator has a legacy reservation established. 0: No initiator has a legacy reservation established.
	NOIV: 1: The following fields are valid. 0: the following fields are not valid and should be ignored.
	Protocol Identifier: 0: Fibre Channel 6: Serial Attached SCSI (SAS)
	Relative Target Port Identifier
	Initiator Identifier: Fibre Channel Port_Name or SAS port identifier
	Persistent Reservations
SET: 1: A persistent reservation is established. 0: No persistent reservation is established.	
PTPL_A: 1: Persist through power loss capability is activated. 0: Persist through power loss capability is not activated.	
Prevent Medium Removal	SET: 1: An initiator has a prevent medium removal established. 0: No initiator has a prevent medium removal established.
	A prevent medium removal descriptor is present for each initiator which has established a prevent medium removal
	Prevent: 00b: Medium removal shall be allowed. 01b: Medium removal shall not be allowed.
	Protocol Identifier: 0: Fibre Channel 6: Serial Attached SCSI (SAS)
	Relative Target Port Identifier
	Initiator Identifier: Fibre Channel Port_Name or SAS port identifier

Clear Nexus Settings

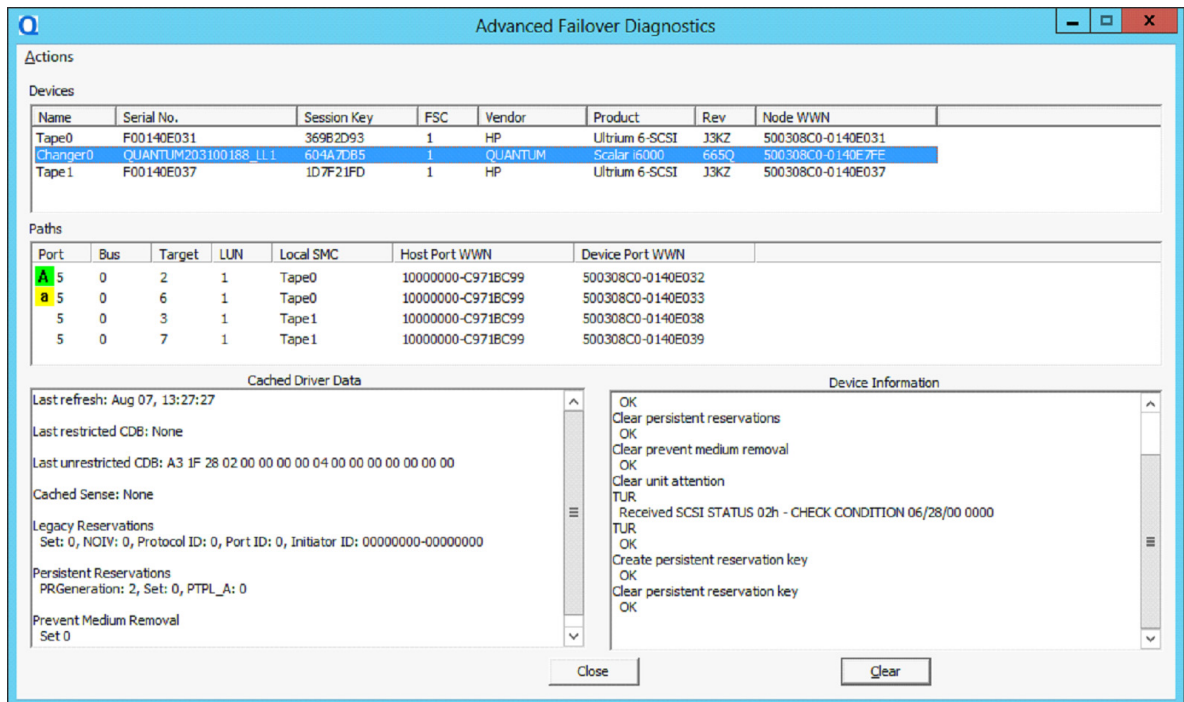
The **Clear Nexus Settings** action is used to clear legacy and persistent reservations placed on a changer device. This can be used to recover from the loss of a host that has placed a reservation on the changer. Before the action is performed, the alert shown in [Figure 11](#) is displayed.

Figure 11 Clear Nexus Settings Warning



The **Device Information** pane in [Figure 12](#) shows the information returned by performing the **Clear Nexus Settings** action. Messages for Verifying session present and Clear legacy reservations have scrolled off the top of the pane.

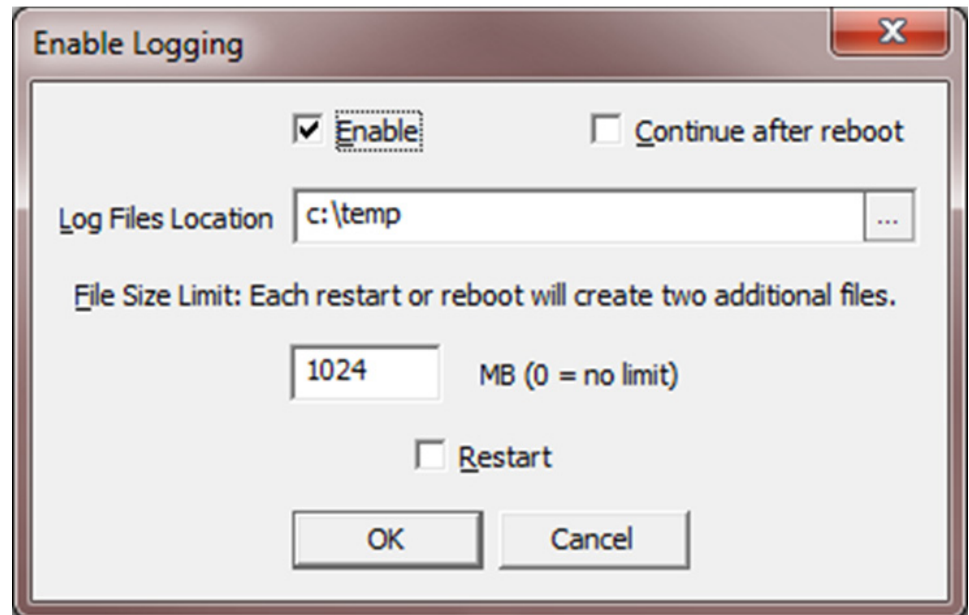
Figure 12 Clear Nexus Settings Messages



Enable Logging

The Enable Logging action provides a mechanism for capturing run-time diagnostic information from the drivers.

Figure 13 Enable Logging



The **Enable** selection will immediately begin capturing driver diagnostic information after **OK** is clicked. The **Enable** setting is not persistent across a system reboot. If the diagnostic information during a reboot is desired then the **Continue after reboot** option must be selected. In the **Log Files Location** field, enter or use the ... button to designate a local directory for log files. There will be two files created when immediately capturing the diagnostic information:

- tapeapfousbf_YYYYMMDDhhmmss.bin
- tapeapfompic_YYYYMMDDhhmmss.bin

where YYYYMMDDhhmmss is the date and time **OK** was clicked. After a reboot there also will be two files in the specified location:

- tapeapfousbf_YYYYMMDDhhmmss.bin.NNN
- tapeapfompic_YYYYMMDDhhmmss.bin.NNN

where, in addition to the date and time, NNNN will be a number from 1 - 16.

The **File Size Limit** value will constrain the size of the log files and if the size limit is reached the oldest log entries are replaced with the new log entries. If 0 is specified as the file size, the log files can become quite large and potentially fill up a hard drive.

When the **Restart** check box is selected and **OK** is clicked, any existing log files are closed and new files are created with updated date and time. Each of the files created after a restart or reboot can grow to the specified size.

Before sending the log files to Quantum, run the application, open the enable logging dialog, deselect **Enable** or select **Restart** and click **OK** to ensure the files have been completely written to disk.

Diagnostic Dumps

There are also additional diagnostic dump logs which can be saved and sent to Quantum for analysis. The first is the standard Windows kernel dump which can be saved when a bug check (blue screen of death) occurs.

Memory Dump Setup

To configure Windows to produce a memory dump, see <http://support.microsoft.com/kb/254649>

The configuration process includes specifying the name and location for the dump file, as well as the type of the dump. The type should be a kernel memory dump, although the larger complete memory dump is acceptable. Do not select the small memory dump. If a bug check occurs, then after a reboot the file can be retrieved and sent for diagnosis.

Bug checks

Following a bug check, restart the system and retrieve the kernel dump for transmission to Quantum. If driver debug dumps were enabled, then retrieve the driver debug log file for transmission to Quantum.

Hangs

In the event the system hangs, the system can be configured to allow forcing a system crash from the keyboard. This is explained in [http://msdn.microsoft.com/en-us/library/ff545499\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff545499(v=VS.85).aspx).

After configuring, if the system hangs then force the crash, reboot, and retrieve the kernel dump and driver debug log for transmission to Quantum

Common Troubleshooting Tasks

There are several troubleshooting tasks which the user can perform to recover from errors:

- Confirming failover operation
- Locating a failed path

Confirming Failover Operation

To confirm that the advanced path failover drivers have installed and are operating correctly, open the system event viewer. See [Windows System Log](#) on page 13 for instructions on how to filter out irrelevant events. The following events should be present; the first event in this list will be the first to occur after system restart:

- **tapeusbf** - Path failover driver loaded
- **tapempic** - Path failover driver loaded

Followed by one instance of the following message for each active path:

- **tapempic** - Path failover session established

For more details on which paths are active and passive, use the diagnostic application (see [Diagnostic Application](#) on page 15).

Locate a Failed Path

The system administrator should have a map of the storage area network which will indicate how ports on the various devices (hosts, switches, and libraries) are connected to one another.

Common causes of failed paths are:

- Loose or unplugged cables
- Loose or unplugged Fibre Channel transceivers
- Failed switch ports
- Powered-off switches
- Failed host bus adapter ports

The first customer-visible indication that a path has failed (e.g., disconnection of a Fibre Channel cable or failure of a switch) will be a support ticket in the library's management GUI. Failure of a single path to a drive should not interfere with operation of an ISV backup application, and so may go unnoticed. However, the problem should be corrected as soon as possible.

Following is a procedure for locating and correcting path failures.

- 1 Examine the support ticket indicating that failover has occurred. The Serial # is that of the failed tape drive and the tape drive coordinates indicate the physical location of the drive. (See Tape Drive Location Coordinates in the library user guide.)
- 2 Use the diagnostic application to confirm the failure.
 - a Launch the application, or if it is already running do **Actions > Refresh Cached Driver Data (F5)**.
 - b In the **Devices** pane, locate the drive whose serial number matches that from the support ticket. Select that device.
 - c In the **Paths** pane, confirm that a port is active.
- 3 Locate the failed connection or SAN component.
 - a Using the location coordinates of the, locate the drive and examine the two Status LEDs labeled **Port 1** and **Port 2**. If either of those LEDs is off, then that is the failed link. The cable may be attached either to an Fibre Channel I/O blade in the library or directly to an external Fibre Channel switch.
 - b If the status LEDs are both on and the library is connected to an external switch, then use the Fibre Channel switch management feature to locate ports that are not functioning.
 - c Examine the Fibre Channel host bus adapters in the host. If one has a light that indicates a malfunction, then that may be the problematic link.

After correcting the connection problem, refresh the diagnostic application to confirm that all of the expected failover paths are available on all devices.

Linux Driver Theory of Operation

Overview

The advanced path failover drivers support Data Path Failover (DPFO) and Control Path Failover (CPFO) on HP LTO-6 drives integrated into Quantum automation. The Linux SCSI driver consists of 3 distinct layers, referred to as the upper, mid, and low-level.

The upper level provides the interface to applications. The primary components of this level are 'st', which provides the interface for tape drives, and 'sg', which provides a generic interface. All tape drives also have a corresponding generic counterpart. This level creates the entries for individual devices in Linux's /dev directory.

The middle level uses the standard Linux block device interface to 'glue' the upper layer to the lower level. Its primary components are `scsi`, `scsi_lib`, and `scsi_mod`. A structure named `scsi_device` contains the information that the middle level uses to route the command to the correct lower level handler.

The lower level provides all of the drivers for individual Host Bus Adapters (HBAs). Normally, each HBA vendor provides a driver for their own hardware.

Different versions of the Linux kernel, even if in the same major kernel version, can have different interfaces between the levels.

The standard Linux kernel will create a new entry in the /dev directory for every path to a device. For example, if there were 4 paths to a particular media changer, you might see /dev/sg0, /dev/sg1, /dev/sg2, and /dev/sg3 all referring to that media changer.

The advanced path failover drivers for Linux replace the normal SCSI Tape and SCSI Generic drivers. The advanced path failover drivers for Linux pass all SCSI commands for devices that do not support advanced path failover through the same code path that is followed when the standard drivers are loaded and route commands for devices that do support failover through the new failover driver.

The failover driver will resolve all paths to a device into a single 'sg' and 'st' entry in the /dev directory for each device. Requests to open or send commands to that device path will use the path selected by the failover driver. Alternate paths that address devices that are already known are entered into an internal alternate path list and device files are not created for those paths.

Driver Components

The failover functionality for Linux is provided by the `pfo` driver and modifications made to the standard `st` and `sg` drivers to call into the `pfo` driver if a device supports advanced path failover. The modified `st` and `sg` drivers are called `stmp` and `sgmp`.

Device Firmware

These three Linux drivers work in conjunction with special firmware running in the changer and tape devices. The special firmware ensures that the host computer and devices stay synchronized on state and position information. The advanced path failover drivers hide the details of this from the rest of the system. The objective is for the rest of the system, including all applications

running on the system and all other device drivers, to see a single changer or tape device but not the redundant paths to the failover-enabled devices. If a path failure occurs it should be transparent to the rest of the system including applications. If a path fails, the drivers work with the device to transfer communication to a new path and recover the command that was in process when the path failed.

Additional Sense Codes

Tape drives supporting advanced path failover use vendor specific SCSI additional sense codes (ASCs) to report certain conditions to the failover drivers. These additional sense codes are handled by the failover drivers in normal operation, and should not be visible to applications (see [Table 2](#) on page 11).

Tape Load Balancing

In order to improve throughput to multiple tape drives the driver will attempt to avoid having an unbalanced number of tape drives using the same HBA path at the same time. The level of path switching activity is adjustable.

The driver supports three levels of load balancing:

- **Full Balancing** - the most comprehensive form of load balancing.
- **Less Balancing** - the least comprehensive form of load balancing. This is the default when the server is booted.
- **No load balancing** - load balancing is not a factor when assigning a drive to an HBA.

The driver will attempt to balance the load each time a device file is opened to a tape device. The normal on-the-fly path failover algorithms will not take other tape drive usage of an HBA into consideration. The usage of the paths will only be evaluated at the next open of the device file.

The path will not be changed to a different path if another thread is holding the `/dev` file open. If multiple threads have the `/dev` file open, the driver will not interfere with the path selection that has been made by another thread.

The load balancing algorithms do not take path speed into consideration when selecting a path; they only attempt to select a sparsely used HBA when a new `/dev` file to a tape drive is opened.

If the load balancing algorithm is unable to select an appropriate path, the path selection algorithm reverts to previous techniques.

The user's selection of a 'preferred' path will take priority over the load balancing method.

Full Balancing algorithm

When a device file is opened to a tape drive, the driver determines the number of open device files to tape drives using the HBA of the last known good path. The driver selects a path on an HBA that is in use by the fewest number of open tape drive device files. It will change paths even if there has not been a path failure event.

Using the Full Balancing setting can result in frequent path changes on a busy system that has multiple drives connected to multiple HBAs. The library is not

always able to distinguish a path change that is the result of a path failure from a path change to optimize data throughput. Each path change can cause a path failover alert from the library. If alerts are enabled in the library and path change alerts are undesirable, the Less Balancing algorithm might be a better choice.

Less Balancing algorithm

When a device file is opened to a tape drive, the driver determines the number of other tape drives using the HBA of the last known good path. The driver selects a path on an HBA that is in use by the fewest number of other tape drives. The driver will not select a different path if there is a failover session already established on that path to that tape drive.

If a path is lost, the driver will attempt to balance traffic among the remaining paths at that time.

This algorithm is not as effective as Full Balancing in optimizing data throughput, but minimizes the frequency of library path failure alerts.

Setting the load balancing level

The level of load balancing can be set on the fly. This setting is volatile across reboot.

To set Full Balancing:

```
echo 2 > /sys/module/pfo/parameters/pfo_balance
```

To set Less Balancing:

```
echo 1 > /sys/module/pfo/parameters/pfo_balance
```

To stop load balancing:

```
echo 0 > /sys/module/pfo/parameters/pfo_balance
```

Setting the load balancing reboot option

The default load balancing algorithm is Less Balancing.

To set Full Balancing at all reboots:

```
echo "options pfo pfo_balance=2" > /etc/modprobe.d/pfo-balance.conf
```

To not use load balancing at all reboots:

```
echo "options pfo pfo_balance=0" > /etc/modprobe.d/pfo-balance.conf
```

Rebalancing the paths

The rebalance command performs a one-time path balancing process, dividing the drives evenly among all HBA paths. This is especially useful when a path returns to service while using the Less Balancing algorithm. This command is likely to cause library path failure events.

The balancing process starts by placing each drive on its preferred path, if one has been assigned. It will then balance the paths for any drives that do not have

a preferred path. The command will balance paths even if there are sessions established with the tape drives.

To balance the tape drive paths:

```
echo rebalance > /sys/bus/scsi/drivers/pfo/ctrl
```

Enable Advanced Path Failover in the Library

Tape libraries that support advanced path failover provide a mechanism in the library user interface for enabling advanced path failover by installing a license. See the library user guide for specific instructions.

Advanced Path Failover Drivers for Linux

Version support

Advanced path failover drivers are available for the following versions of Linux:

Table 8 Release driver installers

Linux Versions	Quantum Installer File
Red Hat/CentOS 6.2, 6.3, 6.4, 6.5 and 6.6	rh6-qtmappf-1.0.9-3.x86_64.rpm

Install Drivers

Make sure that the drivers are for the revision of Linux your server is running. Drivers for a specific revision of Linux might not load in other revisions.

Due to a defect in the **st** driver supplied by Red Hat, the driver does not unload cleanly. If the **st** driver was loaded, it must be blocked and then the system rebooted to unload the **st** driver cleanly. The **st** and **sg** drivers will be properly blocked when the rpm file is installed. If the **st** driver was installed, the system will display a message indicating the need to reboot during the .rpm install process.

Note: This does not apply to red Hat/CentOS 6.5.

To install the drivers:

- 1 Import the Quantum APF GPG public key with the following command:

```
# rpm --import qtmappf-public-GPG-key.asc
```
- 2 Run Quantum Installer:

```
#rpm -ivh rh6-qtmappf-1.0.9-3.x86_64.rpm
```

Note: Installer will check the following:

- Red Hat release version
- Whether or not normal distribution drivers are loaded
- Whether or not normal distribution drivers are blocked

Installer will display the following:

- ERROR: Modules in this .rpm file only support Red Hat/CentOS 6.2 through 6.6
 - ERROR: Could not insert the <module name> module. Please reboot the system to load Quantum Advanced Path Failover modules.
-

3 Check successful installation:

```
#rpm -q qtmapf
```

Upgrade Drivers

The advanced path failover drivers for Linux can be updated to a newer version without rebooting the system and by using the same Quantum Installer:

```
#rpm -Uvh rh6-qtmapf-1.0.9-3.x86_64.rpm
```

Note: Installer will check the following:

- Use count for advanced path failover drivers

Installer will display the following:

- ERROR: Cannot remove the existing <module name> module. The use count is not 0. Close any open /dev/st or dev/sg files, then retry.
-

Uninstall Drivers

The advanced path failover drivers for Linux can be uninstalled using the same Quantum Installer:

```
#rpm -e qtmapf-1.0.9-3.x86_64.rpm
```

Note: Installer will check the following:

- Use count for advanced path failover drivers

Installer will display the following:

- ERROR: Could not remove the old <module name> module! Please reboot the system to unload the <module name> module.
- WARNING: The Advanced Path Failover modules have been removed. There are currently no **ch**, **st** or **sg** modules loaded. Please reboot the system to load the original modules, or execute:

```
modprobe sg  
modprobe st  
modprobe ch
```

Debug Failover Operation

The advanced path failover drivers support several configuration and diagnostic functions via a command line interface. Most of the functions are accessed by direction commands to a specific path. The following example show the command for a specific path and the commands may be directed at any path by replacing the device identifier with the identifier from the **sg** device. For example, a command executed on **pfo3** will report information for the device at **sg3**.

Command Line User Interface

There are several configuration queries which the user can perform:

- View current drivers revision
- View device status
- View cached reservations and prevent/allow settings information
- View connected hosts information

View Current Drivers Revision

The version number can be found by executing `cat /proc/scsi/sg/version`.

There is a single date that is identical for all three drivers (**pfo**, **stmp**, **sgmp**) associated with a single release of a code set. This date can be seen in six places. It is in the **date:** output of the **modinfo** for each driver as well as in the relevant **/sys** file for each driver.

```
cat /sys/bus/scsi/drivers/pfo/version
```

```
cat /sys/bus/scsi/drivers/st/version
```

```
cat /proc/scsi/sg/version
```

```
modinfo pfo.ko
```

```
modinfo stmp.ko
```

```
modinfo sgmp.ko
```

View Device Status

The user can view the status of a device that is controlled by the failover drivers by reading a file in the **/sys** file system. Here is an example of how to see the path status for **/dev/sg3**.

```
cat /sys/class/pfo/pfo3/paths
```

Table 9 Glossary

Term	Description
active_drive	A path to an auto-changer via an advanced path failover active drive.
down	The driver is aware of this path, but if the driver attempted to use this path now it would fail and another path would need to be chosen.
last	The path that the driver was most recently able to successfully perform a SCSI command on. This is probably the path that will be used next. There are no /dev files open to this device.
open	The path that the driver was most recently able to successfully perform a SCSI command on. This is probably the path that will be used next. There is at least one /dev file open to this device.
prefer	The driver will attempt to send the next command on this path after the next device file open() call. This path will override the last known good path.
session_key	The identifier number of the open session, in hexadecimal.
wwnn	World-wide node name of the device.
wwpn	World-wide port name of the FC port on the device.

Example 1 Example Paths

```
# cat /sys/class/pfo/*/paths
sg=/dev/sg0 st=/dev/st0 wwnn=50014382c6c2f001 type=tape
0:0:3:0 up - - wwpn=50014382c6c2f003
0:0:0:0 up - - wwpn=50014382c6c2f002
1:0:0:0 up - - wwpn=50014382c6c2f002
1:0:2:0 up - - wwpn=50014382c6c2f003
sg=/dev/sg1 st=none wwnn=50014382c6c2f800 type=changer
session_key=cae82d14
0:0:3:1 up - - active_drive
0:0:0:1 up last prefer active_drive
0:0:1:1 up - - -
0:0:2:1 up - - -
1:0:0:1 up - - active_drive
1:0:1:1 up - - -
1:0:2:1 up - - active_drive
1:0:3:1 up - - -
sg=/dev/sg2 st=/dev/st1 wwnn=50014382c6c2f007 type=tape
0:0:1:0 up - - wwpn=50014382c6c2f008
0:0:2:0 up open - wwpn=50014382c6c2f009
1:0:1:0 up - - wwpn=50014382c6c2f008
```

1:0:3:0 up - - wwpn=50014382c6c2f009

In this example, the tape drive at the top of the list can be accessed via either `/dev/sg0` or `/dev/st0`. The other tape drive is available via either `/dev/sg2` or `/dev/st1`. An application has a device file open to the drive at the bottom of the list; and the most recent command was sent to the drive via the path 0:0:2:0.

The library is accessed via `/dev/sg1`. The path most recently used to access the library was 0:0:0:1. At some time in the past an administrator specified path 0:0:0:1 to be the “preferred” path to the library.

There are four paths to each drive. Since both drives are connected to the library there appear to be eight paths available to the library. The drive at `/dev/st0` is the “active_drive,” which means that the driver will use that drive to access the library. The active drive can be inferred by looking at the path numbers. The last digit (e.g. the 1 in 0:0:0:1) is the lun number of the device accessed by that path. A drive has a LUN of 0. A library has a LUN of 1. The port names are not reported by the library because the library uses ports on the drives so the port names are not available in the report.

All of the paths show their status as being “up.” This means that the driver believes that all of those paths are currently available for use. The state of “down” is temporary. It is only visible while the path is being destroyed. Once the path becomes completely unusable, the driver removes it from the list. Any path that remains in a “down” state for more than a second should be disconnected and reconnected to reestablish a reliable connection. The path cannot be disconnected via the Linux operating system; disconnect the path by physically disconnecting the FC cable, disabling the port in the FC switch, or power cycling the drive.

View Cached Reservation Information

The advanced path failover drivers will track reservation information for tape libraries to enable rebuilding the reservations if a failure requires changing which drive is hosting the library control path. The following commands will display the reservation and prevent/allow information cache buffers to the `/var/log/messages` file and the console.

Table 10 Reservation and Prevent/Allow Settings

Setting Type	Command
Legacy Reservations	<code>echo legacy > /sys/class/pfo/pfo4/ctrl</code>
Persistent Reservations	<code>echo persistent > /sys/class/pfo/pfo4/ctrl</code>
Prevent/Allow Medium Removal Conditions	<code>echo prevent > /sys/class/pfo/pfo4/ctrl</code>

Clear Cached Reservations and Prevent/Allow Information

The following command will clear all reservations on the Library. The number 4 used in the example below should be replaced with the number of the device you wish to modify.

Table 11 Clear Reservation and Prevent/Allow Settings

Setting Type	Command
Legacy Reservations, Persistent Reservations and Prevent/Allow Medium Removal Conditions	<code>echo clear > /sys/class/pfo/pfo4/ctrl</code>

View Connected Hosts Information

The target device keeps a list of all hosts that are currently connected and the driver can retrieve that list and report all hosts that are connected. Each host connection is called a failover session. Hosts that are disconnected without a clean shutdown (e.g., a path failure or a system crash) will remain in the list of sessions. Thus this list may include connections that are no longer active.

To report all sessions that are connected to a specific device cat the 'sessions' file. Example:

```
cat /sys/class/pfo/pfo3/sessions
```

To report all sessions that are connected to any failover device cat all 'sessions' files. Example:

```
cat /sys/class/pfo/*/sessions
```

Control Path

Each path is assigned its own unique integer number within the advanced path failover drivers. This number is used internally to keep track of relationships between paths and devices. In the console output there are strings "path=" that show this internal value. There is no correlation between advanced path failover path numbers and the Linux path numbers (e.g. 8:0:2:0). To determine which advanced path failover path numbers are associated with which Linux path numbers execute the following command:

```
echo dump > /sys/bus/scsi/drivers/pfo/ctrl
```

The output for this operation can be found on the serial console output and in the /var/log/messages file. Performing this command can provide valuable debug information.

Set Preferred Path

Normally the driver will attempt to use the last known good path. You can override that behavior by selecting a "preferred" path. The preferred path is the path that the driver will attempt to use at the next `open()` of the device file.

The path will not change to the preferred path if there is any other thread holding the /dev file open. If multiple threads have the /dev file open, then the driver will not interfere with the path selection that has been made by another thread.

The path will not change to the preferred path if it is indicating a library control path using a passive drive. The `pfo` driver will select an active drive instead.

You can configure the preferred path by writing to a /sys file. For example, to configure a preferred path on /dev/sg5:

```
echo prefer=8:0:1:0 > /sys/class/pfo/pfo5/ctrl
```

You can disable the configuration of a preferred path by creating an empty prefer entry in the ctrl file. For example:

```
echo prefer > /sys/class/pfo/pfo5/ctrl
```

Path Rotation Function

Path rotation is a test function that will cause the driver to change which path is used every time the device file is closed and reopened. To cause the path to change with every `open()` of the `/dev` file set `rotate=1`. For example:

```
echo rotate=1 > /sys/bus/scsi/drivers/pfo/ctrl
```

To stop path rotation set `rotate=0`. For example:

```
echo rotate=0 > /sys/bus/scsi/drivers/pfo/ctrl
```

Debug Logging Output

There are three means of obtaining diagnostic information about the operation of the advanced path failover mechanism:

- Examine the library tickets.
- Examine the `/var/log/messages` file.
- Examine the trace log.

Library Support Tickets

When the library detects a path failover, it will generate an appropriate support ticket. Tickets are summarized in [Table 4](#) on page 13.

Verbosity Control

The user can control the volume of debug messages being sent to the console and to the file `/var/log/messages`.

The default output level is minimal.

To increase it to the maximum:

```
echo 0xffff > /sys/bus/scsi/drivers/pfo/debug_flag
```

To reduce it to a minimum:

```
echo 0x0000 > /sys/bus/scsi/drivers/pfo/debug_flag
```

Table 12 Verbosity Control

Value	Console and <code>/var/log/messages</code> Output Level	Included into Trace Log
0	Minimum	No
0x1111	ERROR	No
0x3333	WARNING	Yes
0x7777	INFO	Yes
0xffff	DEBUG	Yes

Note: Increased debug logging will affect performance.

To read the current debug flag state:

```
cat /sys/bus/scsi/drivers/pfo/debug_flag
```

or

```
cat /sys/module/pfo/parameters/pfo_debug_flag
```

It is possible to make verbosity control persistent on server reboots. If debugging capabilities need to be automatically enabled at each reboot, it is best to prepare the configurations for this feature before the .rpm install is executed. A new file will need to be created to cause the debug flag to be used automatically upon reboot when the drivers have been installed with an rpm file. This new file tells modprobe to set the debug parameter.

```
echo "options pfo pfo_debug_flag=0xffff" > /etc/modprobe.d/pfo-debug.conf
```

Trace Log

The failover driver keeps an internal buffer of the most recent significant events which may be read following an event. To read the trace log `cat` the 'trace' file. Example:

```
cat /sys/bus/scsi/drivers/pfo/trace
```

Install and Use HP-UX Advanced Path Failover Drivers

Overview

The advanced path failover drivers support Data Path Failover (DPFO) and Control Path Failover (CPFO) on HP LTO-6 drives integrated into Quantum automation.

Advanced path failover for HP-UX is implemented by updating HP-UX drivers to support advanced path failover with the LTO-6 tape drives. The drivers function as both failover and non-failover drivers.

The updated drivers are:

- HP-UX tape driver (estape) - used for data path failover
- HP-UX media changer driver (eschgr) - used for control path failover
- HP-UX SCSI stack driver (esctl) - used for data path and control path failover

During the device open, the device driver checks to see whether the device is capable and enabled for advanced path failover. If so, the device driver opens a failover session and continues with further device operations. Otherwise, the driver uses the non-failover driver code. Note that until the device is opened for the first time, the drivers do not know whether the device is capable of advanced path failover or not.

HP-UX 11i v3 performs automatic load balancing and will open new tape drive connections using the best available path to spread the load across all available

paths. In HP-UX 11i v3, the default load balance policy for tape drives and libraries is the "Path-lock-down" policy. With this policy, the host determines a path for sending I/O requests to the device when the device is opened for the first time and this path remains fixed. Without the advanced path failover feature installed, if this path fails, a new path is only chosen when the device is next opened. With the advanced path failover feature, the best path will be chosen as the lock-down path. If this path fails, the driver will automatically fail over to the next available best path, which becomes the new lock-down path for all further I/O requests.

For additional information see the following man pages: **scsimgr(1M)**, **ioscan(1M)**, **mknod(2)**, **mksf(1M)**, **rmsf(1M)**.

Install Drivers

The patch catalog is available at <http://patch-hub.corp.hp.com/wtec/catalog/>. To locate the patches in the catalog, search for `estape`, `eschgr`, and `esctl`, or the patch number and then look at the `Prepby` field to see if there is a superseding patch.

To install the advanced path failover drivers, use the standard HP-UX kernel patch installation process to install the following patches on the HP-UX host servers running HP-UX 11i v3:

- HP-UX Tape driver patch (`estape`) - PHKL_43680 or superseding patch
- HP-UX Media changer driver patch (`eschgr`) - PHKL_43681 or superseding patch
- HP-UX SCSI stack (Mass storage stack) driver patch (`esctl`) - PHKL_43819 or superseding patch

The server will automatically reboot as part of the installation process.

Commands to View Connected Devices

You can use `ioscan` to view the tape and library (media changer) devices connected to the HP-UX host. The device special file (DSF) is listed as the last item in the description as shown in bold type.

Example 1 Using `ioscan (1M)` to view tape devices

```
# ioscan -knNfC tape
```

```
Class I   H/W Path          Driver  S/W State  H/W Type  Description
=====
tape 18  64000/0xfa00/0xd  estape  CLAIMED    DEVICE     HP Ultrium 6-SCSI
      /dev/rtape/tape18_BEST /dev/rtape/tape18_BESTb /dev/rtape/tape18_BESTn /
      dev/rtape/tape18_BESTnb
tape 20  64000/0xfa00/0xf  estape  CLAIMED    DEVICE     HP Ultrium 6-SCSI
      /dev/rtape/tape20_BEST /dev/rtape/tape20_BESTb /dev/rtape/tape20_BESTn /
      dev/rtape/tape20_BESTnb
```

Example 2 Using `ioscan (1M)` to view library devices

```
# ioscan -knNfC autoch
```

```
Class  I   H/W Path          Driver  S/W State  H/W Type  Description
=====
```

```

autoch 18 64000/0xfa00/0x12  eschgr CLAIMED  DEVICE  Quantum Scalar i6000
/dev/rchgr/autoch18
autoch 19 64000/0xfa00/0x13  eschgr CLAIMED  DEVICE  Quantum Scalar i6000
/dev/rchgr/autoch19
autoch 20 64000/0xfa00/0x14  eschgr CLAIMED  DEVICE  Quantum Scalar i6000
/dev/rchgr/autoch20
autoch 30 64000/0xfa00/0x3e  eschgr CLAIMED  DEVICE  Quantum Scalar i6000
/dev/rchgr/autoch30
48 Installing

```

Example 3 Finding HP-UX paths for drives listed on the library interface

To find this information with HP-UX commands, use `ioscan -knNfC autoch` to see the list of tape libraries, and then use `ioscan -m lun -H <H/W path>` for the partition. For example, if the LUN hardware path for the library is 64000/0xfa00/0x12:

ioscan -m lun -H 64000/0xfa00/0x12

```

Class I   Lun H/W Path      Driver S/W State H/W Type Health Description
=====
autoch18 64000/0xfa00/0x12  eschgr CLAIMED  DEVICE  limited Quantum Scalar
i6000
0/0/0/9/0/0/1.0x50014382c6e4f002.0x1000000000000 <- Passive control path
0/0/0/9/0/0/1.0x50014382c6e4f003.0x10000000000000 <- Passive control path
0/0/0/9/0/0/1.0x50014382c6e4f009.0x10000000000000 <- Active control path
0/0/0/9/0/0/1.0x50014382c6e4f008.0x10000000000000 <- Active control path
0/0/0/9/0/0/0.0x50014382c6e4f003.0x10000000000000 <- Passive control path
0/0/0/9/0/0/0.0x50014382c6e4f009.0x10000000000000 <- Active control path
0/0/0/9/0/0/0.0x50014382c6e4f008.0x10000000000000 <- Active control path
0/0/0/9/0/0/0.0x50014382c6e4f002.0x10000000000000 <- Passive control path
/dev/rchgr/autoch18 <- Device special file for media changer

```

The reported LUN hardware path has three parts, separated by a period. For example, in the path

```
0/0/0/9/0/0/0.0x50014382c6e4f002.0x10000000000000
```

Part 1 . Part 2 . Part 3

- **Part 1:** Hardware path for the HBA. In this example, 0/0/0/9/0/0/0.
- **Part 2:** Target port WWN. In this example, 0x50014382c6e4f002. This is the WWN that is displayed in the library web interface.
- **Part 3:** LUN identifier: In this example, 0x10000000000000.

Similarly, to find the special files for tape drives, use `ioscan -knNfC tape` to list the tape drives and then use `ioscan -m lun -H <H/W path>` to see information for the drive. For example, if the LUN hardware path to the tape drive is 64000/0xfa00/0x37:

ioscan -m lun -H 64000/0xfa00/0x37

```

Class I   Lun H/W Path      Driver S/W State H/W Type Health Description
=====
tape 29 64000/0xfa00/0x37  estape CLAIMED  DEVICE  online HP Ultrium 6-SCSI
0/0/0/9/0/0/1.0x50014382c6e4f009.0x0 <- Tape device special file
0/0/0/9/0/0/1.0x50014382c6e4f008.0x0 <- Tape device special file
0/0/0/9/0/0/0.0x50014382c6e4f009.0x0 <- Tape device special file

```

```
0/0/0/9/0/0/0.0x50014382c6e4f008.0x0 <- Tape device special file  
/dev/rtape/tape29_BEST /dev/rtape/tape29_BESTb /dev/rtape/tape29_BESTn  
/dev/rtape/tape29_BESTnb <- Tape device special files
```

Find the Lock-down Path

The load-balance policy used to route data on multiple paths to a tape drive or library is called the “path-lock-down” policy. Use the `scsimgr_get_info` command to see the current lock-down path for a library. For example:

```
# scsimgr get_info -D /dev/rchgr/autoch35  
STATUS INFORMATION FOR LUN : /dev/rchgr/autoch38  
...  
LUN Path used when policy is path_lockdown = 0/0/0/9/0/0/  
1.0x50014382c6e4f009.0x1000000000000
```

Use the `scsimgr_get_attr` command to see the current lockdown path for a library. For example:

```
# scsimgr get_attr -D /dev/rtape/tape28_BEST  
SCSI ATTRIBUTES FOR LUN : /dev/rtape/tape28_BEST  
...  
name = lpt_lockdown  
current = 0/0/0/9/0/0/0.0x100000e00222a6c1.0x200000000000  
default =  
saved =
```

For additional information, see the HP-UX man pages: `scsimgr (1M)`, `ioscan (1M)`, `mknod (2)`, `mksf (1M)`, `rmsf (1M)`.

Troubleshoot

Advanced path failover errors are logged in the `/var/adm/syslog/syslog.log` file as part of the default SCSI I/O tracing function of HP-UX. You can use standard file viewing commands, including `cat`, `vi`, `dmesg -`, and `tail -f`, to view the `syslog.log` file.

Enable or Disable Advanced Path Failover

Advanced path failover is disabled by default. When advanced path failover is disabled, the driver operates as if the device is not capable of using the advanced path failover feature.

When advanced failover is enabled for the library or tape drive, the device resets itself and must be opened using the device special file before the driver will recognize it as an advanced path failover device and use the failover features of the driver. Opening the device is generally done by the host applications.

You can enable or disable advanced path failover using the library web-based interface. For instructions, see the library user guide.

When advanced path failover is disabled, the passive control paths to the library will go into an error state (`NO_HW`). in the `ioscan (1M)` command output. These stale entries do not affect the function of the library. To clear this errors so the device can be accessed using its device special file:

- 1 On the HP-UX host, run `rmsf -H` on the lunpath hardware paths that are in `NO_HW` state. For example:

```
rmsf -H 0/4/0/0/0/1.0x50014380023560d4.0x1000000000000
```

2 Run `ioscan -kfNH <HBA path>`. For example:

```
ioscan -kfNH 0/4/0/0/0
```

It is recommended to enable or disable advanced path failover when the library is not opened by any applications. If the advanced path failover is disabled while an application is accessing the library, all of the library's lunpaths will go offline and I/O requests to the library will fail.

