Multipath Drivers Guide

Quantum StorNext QS/QD SANtricity Storage Manager 11.20

6-68262-02 Rev A

StorNext Q-Series Storage Failover Drivers Guide, 6-68262-02 Rev A, May 2015, Product of USA.

Table of Contents

Deciding Whether to Use this Guide

The guide describes how to install and configure the supported failover drivers that are used with the storage management software to manage the path control, connection status, and other features of your storage array. Use this guide if you want to accomplish these goals:

- Install a host bus adapter (HBA) driver in failover mode on a system running either Windows, Linux, AIX, or Solaris operating software.
- Configure multiple physical paths (multipaths) to storage and want to follow a standard installation and configuration using best practices.
- This guide does not provide information about device-specific information, all the available configuration options, or a lot of conceptual background for the tasks.

This guide is based on the following assumptions:

- You have the basic configuration information for your storage array and have a basic understanding of path failover.
- Your storage system has been successfully installed.
- Your storage system supports the redundant controller feature.

Where to Find the Latest Information About the Product

You can find information about the latest version of the product, including new features and fixed issues, and a link to the latest documentation at the following address: http://www.quantum.com/.

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments to http://www.quantum.com/. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

For further assistance, or if training is desired, contact the Quantum Customer Support Center:

United States

1-800-284-5101 (toll free)

+1-720-249-5700

EMEA

+800-7826-8888 (toll free)

+49-6131-3241-1164

APAC

+800-7826-8887 (toll free)

+603-7953-3010

Overview of Failover Drivers

Failover drivers provide redundant path management for storage devices and cables in the data path from the host bus adapter to the controller. For example, you can connect two host bus adapters in the system to the redundant controller pair in a storage array, with different bus cables for each controller. If one host bus adapter, one bus cable, or one controller fails, the failover driver automatically reroutes input/output (I/O) to the good path. Failover drivers help the servers to continue to operate without interruption when the path fails.

Failover drivers provide these functions:

- They automatically identify redundant I/O paths.
- They automatically reroute I/O to an alternate controller when a controller fails or all of the data paths to a controller fail.
- They check the state of known paths to the storage array.
- They provide status information on the controller and the bus.
- They check to see if Service mode is enabled on a controller and if the AVT or asymmetric logical unit access (ALUA) mode of operation has changed.
- They provide load balancing between available paths.

Failover Driver Setup Considerations

Most storage arrays contain two controllers that are set up as redundant controllers. If one controller fails, the other controller in the pair takes over the functions of the failed controller, and the storage array continues to process data. You can then replace the failed controller and resume normal operation. You do not need to shut down the storage array to perform this task.

The redundant controller feature is managed by the failover driver software, which controls data flow to the controller pairs. This software tracks the current status of the connections and can perform the switch-over.

Whether your storage arrays have the redundant controller feature depends on a number of items:

- Whether the hardware supports it. Refer to the hardware documentation for your storage arrays to determine whether the hardware supports redundant controllers.
- Whether your OS supports certain failover drivers. Refer to the installation and support guide for your OS to determine if your OS supports redundant controllers.
- How the storage arrays are connected.

With the I/O Shipping feature, a storage array can service I/O requests through either controller in a duplex configuration. However, I/O shipping alone does not guarantee that I/O is routed to the optimized path. With Windows, Linux and VMWare, your storage array supports an extension to ALUA to address this problem so that volumes are accessed through the optimized path unless that path fails. With SANtricity® Storage Manager 10.86 and subsequent releases, Windows and Linux device-mapper multipath (DM-MP) have I/O shipping enabled by default.

Supported Multipath Drivers

The information in this table is intended to provide general guidelines. Please refer to the interoperability matrix for compatibility information for specific HBA, Multipath driver, OS level, and base system support.

Table 1. Matrix of Supported Multipath Drivers

Operating System	Multipath Driver	Recommended Host Type	Default Host Type selected by Host Context Agent[1]
Windows Server	MPIO with NetApp E-Series DSM (with ALUA support)	Windows or Window Clustered	Windows or Windows Clustered
Windows	ATTO with TPGS/ALUA	Windows ATTO **NOTE** You must use ATTO FC HBAs.	Windows or Windows Clustered
Linux	NetApp MPP/RDAC	Linux (MPP/RDAC)	Linux (MPP/RDAC)
Linux	DMMP with RDAC handler (with ALUA support)	Linux (DM-MP)	Linux (MPP/RDAC)
Linux	ATTO with TPGS/ALUA	Linux (ATTO) **NOTE** You must use ATTO FC HBAs.	Linux (MPP/RDAC)
Solaris	MPxIO (non-TPGS)	Solaris Version 10 or earlier	Solaris (version 10 or earlier)
Solaris	MPxIO (TPGS/ALUA)	Solaris Version 11 or later	Solaris (version 10 or earlier)
HP-UX	Native TPGS/ALUA	HP-UX	HP-UX
VMWare	Native VMWare with VMW_SATP_ALUA NMP plug-in	VMWare	N/A
Mac	ATTO with TPGS/ALUA	Mac OS	N/A
AIX VIOS	Native MPIO	AIX MPIO	AIX MPIO
ONTAP	Native RDAC	Data ONTAP (RDAC)	N/A
ONTAP	Native ALUA	Data ONTAP (ALUA)	N/A
Linux	VxDMP	Linux (Symantec Storage Foundation)	N/A

[1] The host context agent is part of the **`SMagent`** package and is installed with SANtricity Storage Manager. After the host context agent is installed and the storage is attached to the host, the host context agent sends the host topology to the storage controllers through the I/O path. Based on the host topology, the storage controllers will automatically define the host and the associated host ports, and set the host type. The host context agent will send the host topology to the storage controllers only once and any subsequent changes made in SANtricity Storage Manager will be persisted.

NOTE If the host context agent does not select the recommended host type, you must manually set the host type in SANtricity Storage Manager. To manually set the host type, from the Array Management Window, select the **Host Mappings** tab and select the host, then select **Host Mappings** >**Host** >**Change Host Operating System**. If you are not using partitions (for example, no Hosts defined), set the appropriate host type for the Default Group by selecting **Host Mappings** > **Default Group** > **Change Default Host Operating System**.

Failover Configuration Diagrams

You can configure failover in several ways. Each configuration has its own advantages and disadvantages. This section describes these configurations:

- Single-host configuration
- Multi-host configuration

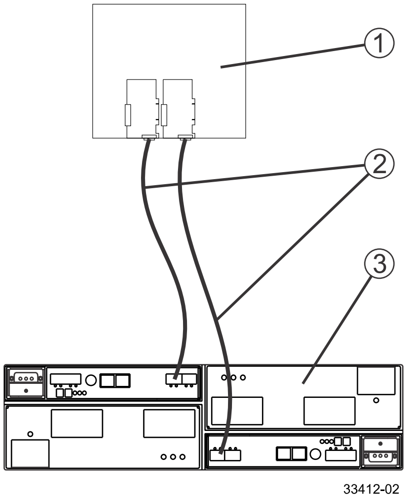
This section also describes how the storage management software supports redundant controllers.

Single-Host Configuration

In a single-host configuration, the host system contains two host bus adapters (HBAs), with each HBA connected to one of the controllers in the storage array. The storage management software is installed on the host. The two connections are required for maximum failover support for redundant controllers.

Although you can have a single controller in a storage array or a host that has only one HBA port, you do not have complete failover data path protection with either of those configurations. The cable and the HBA become a single point of failure, and any data path failure could result in unpredictable effects on the host system. For the greatest level of I/O protection, provide each controller in a storage array with its own connection to a separate HBA in the host system.

Figure 1. Single-Host-to-Storage Array Configuration



1. Host System with Two Fibre Channel Host Bus Adapters
2. Fibre Channel Connection – Fibre Channel Connection Might Contain One or More Switches
3. Storage Array with Two Controllers

Host Clustering Configurations

In a clustering configuration, two host systems are each connected by two connections to both of the controllers in a storage array. SANtricity Storage Manager, including failover driver support, is installed on each host.

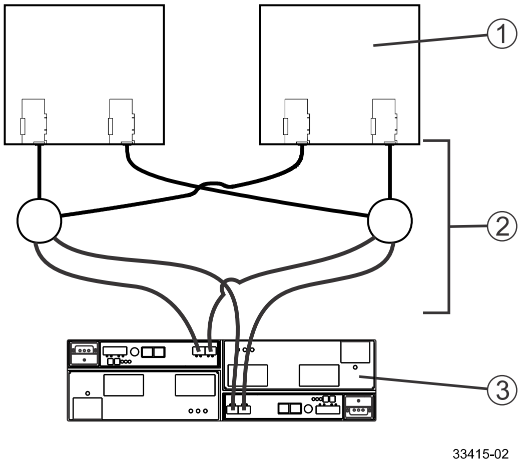
Not every operating system supports this configuration. Consult the restrictions in the installation and support guide specific to your operating system for more information. Also, the host systems must be able to handle the multi-host configuration. Refer to the applicable hardware documentation.

In a clustering configuration, each host has visibility to both controllers, all data connections, and all configured volumes in a storage array.

The following items apply to these clustering configurations:

- Both hosts must have the same operating system version installed.
- The failover driver configuration might require tuning.
- A host system might have a specified volume or volume group reserved, which means that only that host system can perform operations on the reserved volume or volume group.

Figure 2. Multi-Host-to-Storage Array Configuration

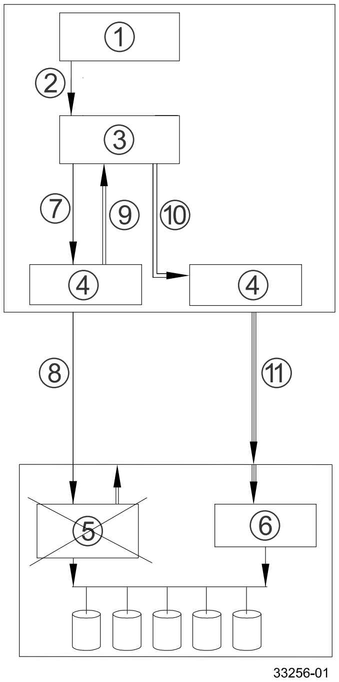


1. Two Host Systems, Each with Two Fibre Channel Host Bus Adapters
2. Fibre Channel Connections with Two Switches (Might Contain Different Switch Configurations)
3. Storage Array with Two Fibre Channel Controllers

Supporting Redundant Controllers

The following figure shows how failover drivers provide redundancy when the host application generates a request for I/O to controller A, but controller A fails. Use the numbered information to trace the I/O data path.

Figure 3. Example of Failover I/O Data Path Redundancy



1. Host Application
2. I/O Request
3. Failover Driver
4. Host Bus Adapters
5. Controller A Failure
6. Controller B
7. Initial Request to the HBA
8. Initial Request to the Controller Failed
9. Request Returns to the Failover Driver
10. Failover Occurs and I/O Transfers to Another Controller
11. I/O Request Re-sent to Controller B

How a Failover Driver Responds to a Data Path Failure

One of the primary functions of the failover feature is to provide path management. Failover drivers monitor the data path for devices that are not working correctly or for multiple link errors. If a failover driver detects either of these conditions, the failover driver automatically performs these steps:

- The failover driver checks for the redundant controller.
- The failover driver performs a path failure if alternate paths to the same controller are available. If all of the paths to a controller are marked offline, the failover driver performs a controller failure. The failover driver provides notification of an error through the OS error log facility.
- The failover driver transfers volume ownership to the other controller and routes all I/O to the remaining active controller.

User Responses to a Data Path Failure

Use the Major Event Log (MEL) to troubleshoot a data path failure. The information in the MEL provides the answers to these questions:

- What is the source of the error?
- What is required to fix the error, such as replacement parts or diagnostics?

Under most circumstances, contact your Technical Support Representative any time a path fails and the storage array notifies you of the failure. Use the Major Event Log to diagnose and fix the problem, if possible. If your controller has failed and your storage array has customer-replaceable controllers, replace the failed controller. Follow the manufacturer’s instructions for how to replace a failed controller.

Dividing I/O Activity Between Two RAID Controllers to Obtain the Best Performance

For the best performance of a redundant controller system, use the storage management software to divide I/O activity between the two RAID controllers in the storage array. You can use either the graphical user interface (GUI) or the command line interface (CLI).

To use the GUI to divide I/O activity between two RAID controllers, perform one of these steps:

- **Specify the owner of the preferred controller of an existing volume** – Select **Volume >> Change >> Ownership/Preferred Path** in the Array Management Window.

NOTE You also can use this method to change the preferred path and ownership of all volumes in a volume group at the same time.

- **Specify the owner of the preferred controller of a volume when you are creating the volume** – Select **Volume >> Create** in the Array Management Window.

To use the CLI, go to the "Create RAID Volume (Free Extent Based Select)" online help topic for the command syntax and description.

Failover Drivers for the Windows Operating System

The failover driver for hosts with Microsoft Windows operating systems is Microsoft Multipath I/O (MPIO) with a Device Specific Module (DSM) for SANtricity Storage Manager.

Terminology

The Device Specific Module (DSM) for SANtricity Storage Manager uses a generic data model to represent storage instances and uses the following terminology.

- **`DeviceInfo`** - A specific instance of a logical unit mapped from a storage array to the host that is visible on an I-T nexus.
- **`MultipathDevice`** - An aggregation of all **`DeviceInfo`** instances that belong to the same logical unit. Sometimes known as a Pseudo-Lun or Virtual Lun.
- **`TargetPort`** - A SCSI target device object that represents a connection between the initiator and target (for example, an I-T nexus). This is also known as a Path.
- **`TargetPortGroup`** - A set of **`TargetPort`** objects that have the same state and transition from state to state in unison. All **`TargetPort`** objects associated with a storage arraycontroller belong to the same **`TargetPortGroup`**, so a **`TargetPortGroup`** instance can be thought of as representing a Controller.
- **`OwningPortGroup`** - The **`TargetPortGroup`** currently being used to process I/O requests for a multi-path device.
- **`PreferredPortGroup`** - The **`TargetPortGroup`** that is preferred for processing I/O requests to a multi-path device. The Preferred Port Group and Owning Port Group may be the same or different, depending on the current context. Preferred Port Groups allow for load balancing of multi-path devices across **`TargetPortGroups`**.
- **`PortGroupTransfer`** - One or more actions that are necessary to switch the Owning Port Group to another **`TargetPortGroup`**, for example, to perform failover of one or more LUNs. (Also known as LUN Transfer or Transfer).

Operational Behavior

System Environment

Microsoft MPIO (MPIO) is a feature that provides multipath IO support for Windows Operating Systems. It handles OS-specific details necessary for proper discovery and aggregation of all paths exposed by a storage array to a host system. This support relies on built-in or third-party drivers called Device-Specific Modules (DSMs) to handle details of path management such as load balance policies, IO error handling, failover, and management of the DSM.

A disk device is visible to two adapters. Each adapter has its own device stack and presents an instance of the disk device to the port driver (storport.sys), which creates a device stack for each instance of the disk. The MS disk driver (msdisk.sys) assumes responsibility for claiming ownership of the disk device instances and creates a multipath device. It also determines the correct DSM to use for managing paths to the device. The MPIO driver (mpio.sys) manages the connections between the host and the device including power management and PnP management, and acts as a virtual adapter for the multipath devices created by the disk driver.

Failover Methods (LUN Transfer Methods)

The DSM driver supports several different command types ("Methods") of Failover that are described in the next sections.

Mode Select

Mode Select provides a vendor-unique request for an initiator to specify which TargetPortGroup should be considered the Owning Port Group.

Target Port Group Support (TPGS)

TPGS provides a standards-based method for monitoring and managing multiple I/O TargetPorts between an initiator and a target. It manages target port states with respect to accessing a DeviceInfo. A given TargetPort can be in different TPGS states for different DeviceInfos. Sets of TargetPorts that have the same state and that transition from state-to-state in unison can be defined as being in the same TargetPortGroup. The following TPGS states are supported.

- **ACTIVE/OPTIMIZED** — TargetPortGroup is available for Read/Write I/O access with optimal performance. This is similar to the concept of a current owning controller.
- **ACTIVE/NON-OPTIMIZED** — TargetPortGroup is available for Read/Write I/O access, but with less than optimal performance.
- **STANDBY** — TargetPortGroup is not available for R/W I/O access, but in the event of losing paths to the active TargetPortGroup, this TargetPortGroup can be made available for Read/Write I/O access. This is equivalent to the concept of a non-owning controller.
- **UNAVAILABLE** — TargetPortGroup is not available for Read/Write I/O access and it might not be possible to transition it to a non-UNAVAILABLE state. An example is a hardware failure.

TPGS support is determined by examining the 'TPGS' field returned from a SCSI INQUIRY request.

Failover Mode

Selective Lun Transfers

Selective LUN Transfer is a failover mode that limits the conditions under which the Owning Port Group for a MultipathDevice is transferred between TargetPortGroups to one of the following cases:

- Transfer the MultipathDevice when the DSM discovers the first TargetPort to the Preferred Port Group.
- Transfer the MultipathDevice when the Owning and Preferred Port Group are the same, but the DSM does not have visibility to those groups.
- Transfer the MultipathDevice when the DSM has visibility to the Preferred Port Group but not the Owning Port Group.

For the second and third case, configurable parameters exist to define the failover behavior. These parameters are described in the Configuration Parameters section.

Failover Method Precedence

The Failover method is determined by the DSM on a storage array-by-storage array basis and is based on a system of precedence as described in the following table.

Table 2. Failover Method Precedence

Failover Method	Precedence	Comments
Forced Use of Mode Select	1	Determined by the `AlwaysUseLegacyLunFailover` configurable parameter. Used when issues are found with TPGS support.
TPGS	2	Determined through a standard Inquiry request.
ModeSelect	3	Default method if all other precedences are invalidated.

ALUA (I/O Shipping)

I/O Shipping is a feature that sends the Host I/O to a MultipathDevice to any Port Group within the storage array. If Host I/O is sent to the Owning Port Group, there is no change in existing functionality. If Host I/O is sent to the Non-Owning Port Group, the firmware uses the back-end storage array channels to send the I/O to Owning Port Group. The DSM driver attempts to keep I/O routed to the Owning Port Group whenever possible.

With I/O Shipping enabled, most error conditions that require failover results in the DSM performing a simple re-route of the I/O to another eligible Port Group. There are, however, cases where failover using one of the Failover Methods previously described are used:

- Moving the MultipathDevice when the DSM discovers the first TargetPort to the Preferred Port Group. This is the failback behavior of Selective LUN Transfer.
- If the ControllerIoWaitTime is exceeded.

When you install or update the software to SANtricity version 10.83 or later, and install or update the controller firmware to version 7.83 or later, support for ALUA is enabled by default.

Path Selection (Load Balancing)

Path selection refers to selecting a TargetPort to a MultipathDevice. When the DSM driver receives a new I/O to process, it begins path selection by trying to find a TargetPort to the Owning Port Group. If a TargetPort to the Owning Port Group cannot be found, and ALUA is not enabled, the DSM driver arranges for MultipathDevice ownership to transfer (or failover) to an alternate TargetPortGroup. The method used to transfer ownership is based on the Failover method defined for the MultipathDevice. When multiple TargetPort's to a MultipathDevice exist, the system uses a load balance policy to determine which TargetPort to use.

Round-Robin with Subset

The Round-Robin with Subset policy selects the most eligible TargetPort in the sequence. TargetPort eligibility is based on a system of precedence, which is a function of DeviceInfo and TargetPortGroup state.

Table 3. Round-Robin with Subset Path Precedence

TargetPortGroup State	Precedence
ACTIVE/OPTIMIZED	1
ACTIVE/NON-OPTIMIZED	2
UNAVAILABLE	3

TargetPortGroup State	Precedence
Any other state	Ineligible

Least Queue Depth

The Least Queue Depth policy selects the most eligible TargetPort with the least number of outstanding I/O requests queued. TargetPort eligibility is based on a system of precedence, which is a function of DeviceInfo and TargetPortGroup state. The type of request or number of blocks associated with the request are not considered by the Least Queue Depth policy.

Table 4. Least Queue Depth Path Precedence

TargetPortGroup State	Precedence
ACTIVE/OPTIMIZED	1
ACTIVE/NON-OPTIMIZED	2
UNAVAILABLE	3
Any other state	Ineligible

Failover Only

The Failover Only policy selects the most eligible TargetPort based on a system of precedence, which is a function of DeviceInfo and TargetPortGroup state. When a TargetPort is selected, it is used for subsequent I/O requests until its state transitions, at which time another TargetPort is selected.

Table 5. Failover Only Path Precedence

TargetPortGroup State	Precedence
ACTIVE/OPTIMIZED	1
ACTIVE/NON-OPTIMIZED	2
UNAVAILABLE	3
Any other state	Ineligible

Least Path Weight

The Least Path Weight policy selects the most eligible TargetPort based on a system of precedence in which a weight factor is assigned to each TargetPort to a DeviceInfo. I/O requests are routed to the lowest weight TargetPort of the Owning Port Group. If the weight factor is the same between TargetPorts then the Round-Robin load balance policy is used to route I/O requests.

Table 6. Least Path Weight Path Precedence

TargetPortGroup State	Precedence
ACTIVE/OPTIMIZED	1
ACTIVE/NON-OPTIMIZED	2
UNAVAILABLE	3
Any other state	Ineligible

Additional Notes On Path Selection

If the only eligible TargetPortGroup states are STANDBY, a Failover Method is initiated to bring the TargetPortGroup state to ACTIVE/OPTIMIZED or ACTIVE/NON-OPTIMIZED.

Online/Offline Path States

The ACTIVE/OPTIMIZED and ACTIVE/NON-OPTIMIZED states reported by TargetPortGroup and DeviceInfo objects are from the perspective of the target (storage array). These states do not take into account the overall condition of the TargetPort connections that exist between the initiator and target. For example, a faulty cable or connection might cause many retransmissions of packets at a protocol level, or the target itself might be experiencing high levels of I/O stress. Conditions like these can cause delays in processing or completing I/O requests sent by applications, and does not cause OS-level enumeration activities (- PnP) to be triggered.

The DSM supports the ability to place the DeviceInfo objects that are associated with a TargetPort into an OFFLINE state. An OFFLINE state prevents any I/O requests from being routed to a TargetPort regardless of the actual state of the connection. The OFFLINE state can be performed automatically based on feature-specific criteria (such as Path Congestion Detection). It also qcan be performed through the multipath utility (dsmUtil) but known as ADMIN_OFFLINE instead. A TargetPort in an ADMIN_OFFLINE state can be placed only in an ONLINE state by an Admin action, host reboot, or PnP removal/add.

Path Congestion Detection

Path Congestion Detection monitors the I/O latency of requests to each TargetPort, and is based on a set of criteria that automatically place the TargetPort into an OFFLINE state. The criteria are defined through configurable parameters, which are described in the Configuration Parameters section.

Example Configuration Settings for the Path Congestion Detection Feature

NOTE Before you can enable path congestion detection, you must set the`CongestionResponseTime`, `CongestionTimeFrame`, and `CongestionSamplingInterval` parameters to valid values.

To set the path congestion I/O response time to 10 seconds do the following:

```
dsmUtil -o CongestionResponseTime=10,SaveSettings
```

To set the path congestion sampling interval to one minute do the following:

```
dsmUtil -o CongestionSamplingInterval=60,SaveSettings
```

To enable Path Congestion Detection do the following:

```
dsmUtil -o CongestionDetectionEnabled=0x1,SaveSettings
```

To set a path to Admin Offline do the following:

```
dsmUtil -o SetPathOffline=0x77070001
```

NOTE You can find the path ID (in this example 0x77070001) using the **`dsmUtil -g`** command.

To set a path Online do the following:

```
dsmUtil -o SetPathOnline=0x77070001
```

Per-Protocol I/O Timeouts

The MS Disk driver must assign an initial I/O timeout value for every non-pass-through request. By default, the timeout value is 10 seconds, although you can override it using the Registry setting called TimeOutValue. The timeout value is considered global to all storage that the MS Disk driver manages.

The DSM can adjust the I/O timeout value of Read/Write requests (those requests passed by MPIO into the DsmLBGetPath() routine) based on the protocol of the TargetPort chosen for the I/O request.

The timeout value for a protocol is defined through configurable parameters, which are described in the Configurable Parameters section.

Wait Times

A Wait Time is an elapsed time period that, when expired or exceeded, causes one or more actions to take place. There is no requirement that a resource, such as a kernel timer, manage the time period which would immediately cause execution of the action(s). For example, an I/O Wait Time will establish a start time when the I/O request is first delivered to the DSM driver. The end time establishes when the I/O request is returned. If the time period is exceeded, an action such as Failover, is initiated between TargetPortGroups.

All Wait Times defined by the DSM driver are configurable and contain the term "WaitTime" as part of the configuration name. Refer to the Configurable Parameters section for a complete list of Wait Times.

SCSI Reservations

Windows Server Failover Cluster (WSFC) uses SCSI-3 Reservations, otherwise known as Persistent Reservations (PR), to maintain resource ownership on a node. The DSM is required to perform some special processing of PR’s because WSFC is not multipath-aware.

Native SCSI-3 Persistent Reservations

Windows Server 2008 introduced a change to the reservation mechanism used by the Clustering solution. Instead of using SCSI-2 reservations, Clustering uses SCSI-3 Persistent Reservations, which removes the need for the DSM to handle translations. Even so, some special handling is required for certain PR requests because Cluster itself has no knowledge of the underlying TargetPorts for a MultipathDevice.

Special Circumstances For Array Brownout Conditions

Depending on how long a brownout condition lasts, Persistent Registration information for volumes might be lost. By design, WSFC periodically polls the cluster storage to determine the overall health and availability of the resources. One action performed during this polling is a PRIN READ KEYS request, which returns registration information. Because a brownout can cause blank information to be returned, WSFC interprets this as a loss of access to the disk resource and attempts recovery by first failing the resource and then performing a new arbitration. The arbitration recovery process happens almost immediately after the resource is failed. This situation, along with the PnP timing issue, can result in a failed recovery attempt. You can modify the timing of the recovery process by using the **`cluster.exe`** command-line tool.

Another option takes advantage of the Active Persist Through Power Loss (APTPL) feature found in Persistent Reservations, which ensures that the registration information persists through brownout or other conditions related to a power failure. APTPL is enabled when a PR REGISTRATION is initially made to the disk resource. You must set this option before PR registration occurs. If you set this option after a PR registration occurs, take the disk resource offline and then bring it back online.

WSFC does not use the APTPL feature but a configurable option is provided in the DSM to enable this feature when a registration is made through the multipath utility. Refer to the Configuration Parameters section for more details.

NOTE The SCSI specification does not provide a means for the initiator to query the target to determine the current APTPL setting. Therefore, any output generated by the multipath utility might not reflect the actual setting.

Auto Failback

Auto Failback ensures that a MultipathDevice is owned by the Preferred TargetPortGroup. It uses the Selective LUN Transfer failover mode to determine when it is appropriate to move a MultipathDevice to its Preferred TargetPortGroup. Auto Failback also occurs if the TargetPorts belonging to the Preferred TargetPortGroup is transitioned from an ADMIN_OFFLINE state or OFFLINE state to an ONLINE state.

MPIO Pass-Through

One of MPIO's main responsibilities is to aggregate all DeviceInfo objects into a MultipathDevice, based partially on input from the DSM. By default, the TargetPort chosen for an I/O request is based on current Load Balance Policy. If an application wants to override this behavior and send the request to a specific TargetPort, it must do so using an MPIO pass-through command (**`MPIO_PASS_THROUGH_PATH`**). This is a special IOCTL with information about which TargetPort to use. A TargetPort can be chosen through one of two of the following methods:

- **PathId** — A Path Identifier, returned to MPIO by the DSM when **`DsmSetPath()`** is called during PnP Device Discovery.
- **SCSI Address** — A SCSI_ADDRESS structure, supplied with the appropriate Bus, Target, and Id information.

Administrative and Configuration Interfaces

This section describes the Windows Management Instrumentation (WMI) and CLI interfaces.

Windows Management Instrumentation (WMI)

Windows Management Instrumentation (WMI) is used to manage and monitor Device-Specific Modules (DSMs).

During initialization, the DSM passes WMI entry points and MOF class GUID information to MPIO, which publishes the information to WMI. When MPIO receives a WMI request, it evaluates the embedded GUID information to determine whether to forward the request to the DSM or to keep it with MPIO.

For DSM-defined classes, the appropriate entry point is invoked. MPIO also publishes several MOF classes that the DSM is expected to handle. MOF classes also can have Methods associated with them that can be used to perform the appropriate processing task.

CLI Interfaces

Multipath Utility (dsmUtil)

The dsmUtil utility is used with the DSM driver to perform various functions provided by the driver.

Configurable Parameters

The DSM driver contains field-configurable parameters that affect its configuration and behavior. You can set these parameters using the multipath utility (dsmUtil). Some of these parameters also can be set through interfaces provided by Microsoft.

Persistence of Configurable Parameters

Each configuration parameter defined by the DSM has a default value that is hard-coded into the driver source. This default value allows for cases where a particular parameter may have no meaning for a particular customer configuration, or a parameter that needs to assume a default behavior for legacy support purposes, without the need to explicitly define it in non-volatile storage (registry). If a parameter is defined in the registry, the DSM uses that value rather than the hard-coded default.

There might be cases where you might want to modify a configurable parameter, but only temporarily. If the host is subsequently rebooted, the value in non-volatile storage is used. By default, any configurable parameter changed by the multipath utility only affects the in-memory representation. The multipath utility can optionally save the changed value to non-volatile storage through an additional command-line argument.

Scope of Configurable Parameters

A localized configurable parameter is one that can be applied at a scope other than global. Currently the only localized parameter is for load balance policy.

Configurable Parameters - Error Recovery

Table 7. Configurable Parameters - Error Recovery

Configuration Parameter	Description	Values
ControllerIoWaitTime	Length of time (sec.) a request is attempted to a controller before failed over.	Min: 0xA Max: 0x12C Default: 0x78 Configured: 0x78
NsdIORetryDelay	Specifies the length of time (in seconds) an I/O request is delayed before it is retried, when the DSM has detected the MPIODisk no longer has any available paths.	Min: 0x0 Max: 0x3C Default: 0x5 Configured: 0x5

Configuration Parameter	Description	Values
IORetryDelay	Specifies the length of time (in seconds) an I/O request is delayed before it is retried, when various “busy” conditions (for example, Not Ready) or an RPTG request needs to be sent.	Min: 0x0 Max: 0x3C Default: 0x2 Configured: 0x2
SyncIoRetryDelay	Specifies the length of time (in seconds) a DSM-internally-generated request is delayed before it is retried, when various "busy" conditions (ex. Not Ready) is detected.	Min: 0x0 Max: 0x3C Default: 0x2 Configured: 0x2

Configurable Parameters - Private Worker Thread Management

Table 8. Configurable Parameters - Private Worker Thread Management

Configuration Parameter	Description	Values
MaxNumberOfWorkerThreads	Specifies the maximum number of private worker threads that will be created by the driver, whether resident or non-resident. If the value is set to zero, then the private worker thread management is disabled.	Min: 0x0 Max: 0x10 Default: 0x10 Configured: 0x10
NumberOfResidentWorkerThreads	Specifies the number of private worker threads created by the driver,. Formally kn.own as NumberOfResidentThreads.	Min: 0x0 Max: 0x10 Default: 0x10 Configured: 0x10

Configurable Parameters - Path Congestion Detection

Table 9. Configurable Parameters - Path Congestion Detection

Configuration Parameter	Description	Values
CongestionDetectionEnabled	A boolean value that determines whether PCD is enabled.	Min: 0x0 (off) Max: 0x1 (on) Default: 0x0 Configured: 0x0
CongestionTakeLastPathOffline	A boolean value that determines whether the DSM driver takes the last path available to the storage array offline if the congestion thresholds have been exceeded.	Min: 0x0 (no) Max: 0x1 (yes) Default: 0x0 Configured: 0x0
CongestionResponseTime	Represents an average response time (in seconds) allowed for an I/O request. If the value of the `CongestionIoCount` parameter is non-zero, this parameter is the absolute time allowed for an I/O request.	Min: 0x1 Max: 0x10000 Default: 0x0 Configured: 0x0

Configuration Parameter	Description	Values
CongestionIoCount	The number of I/O requests that have exceeded the value of the `CongestionResponseTime` parameter within the value of the `CongestionTimeFrame` parameter.	Min: 0x0 Max: 0x10000 Default: 0x0 Configured: 0x0
CongestionTimeFrame	A sliding windows that defines the time period that is evaluated in seconds.	Min: 0x1 Max: 0x1C20 Default: 0x0 Configured: 0x0
CongestionSamplingInterval	The number of I/O requests that must be sent to a path before the <n> request is used in the average response time calculation. For example, if this parameter is set to 100, every 100th request sent to a path will be used in the average response time calculation.	Min: 0x1 Max: 0xFFFFFFFF Default: 0x0 Configured: 0x0
CongestionMinPopulationSize	The number of sampled I/O requests that must be collected before the average response time is calculated.	Min: 0x0 Max: 0xFFFFFFFF Default: 0x0 Configured: 0x0
CongestionTakePathsOffline	A boolean value that determines whether any paths will be taken offline when the configured path congestion thresholds are exceeded.	Min: 0x0 (no) Max: 0x1 (yes) Default: 0x0 Configured: 0x0

Configurable Parameters - Failover Management: Legacy Mode

Table 10. Configurable Parameters - Failover Management: Legacy Mode

Configuration Parameter	Description	Values
AlwaysUseLegacyLunFailover	Boolean setting that controls whether Legacy Failover is used for all Failover attempts, regardless of whether the storage array supports TPGS.	Min: 0x0 Max: 0x1 Default: 0x0 Configured: 0x0
LunFailoverInterval	Length of time (sec) between a Failover event being triggered and the initial failover request being sent to the storage array. Formally known as 'LunFailoverDelay'.	Min: 0x0 Max: 0x3 Default: 0x3 Configured: 0x3
RetryLunFailoverInterval	Length of time (sec) between additional Failover attempts, if the initial failover request fails. Formally known as 'RetryFailoverDelay'.	Min: 0x0 Max: 0x3 Default: 0x3 Configured: 0x3

Configuration Parameter	Description	Values
LunFailoverWaitTime	Length of time (sec) a failover request is attempted for a lun (or batch processing of luns) before returning an error. Formally known at 'MaxArrayFailoverLength'.	Min: 0xB4 Max: 0x258 Default: 0x12C Configured: 0x12C
LunFailoverQuiescenceTime	Length of time (sec) to set in the 'QuiescenceTimeout' field of a Legacy Failover request.	Min: 0x1 Max: 0x1E Default: 0x5 Configured: 0x5
MaxTimeSinceLastModeSense	The maximum amount of time (sec) that cached information regarding TargetPort and TargetPortGroup is allowed to remain stale.	Min: 0x0 Max: 0x60 Default: 0x5 Configured: 0x5

Configurable Parameters - MPIO-Specific

Table 11. Configurable Parameters - MPIO-Specific

Configuration Parameter	Description	Values
RetryInterval	Delay (sec) until a retried request is dispatched by MPIO to the target. Already provided by MPIO, but can be modified.	Min: 0x0 Max: 0xFFFFFFFF Default: 0x0 Configured: 0x0
PDORemovePeriod	Length of time (sec) an MPIO Pseudo-Lun remains after all I-T nexus connections have been lost. Already provided by MPIO, but can be modified.	Min: 0x0 Max: 0xFFFFFFFF Default 0x14 Configured:

Configurable Parameters - Per-Protocol I/O Timeouts

Table 12. Configurable Parameters - Per-Protocol I/O Timeouts

Configuration Parameter	Description	Values
FCTimeOutValue	Timeout value (sec) to apply to Read/Write requests going to FC-based I-T nexus. If set to zero, the timeout value is not changed.	Min: 0x1 Max: 0xFFFF Default: 0x3C Configured: 0x3C
SASTimeOutValue	Timeout value (sec) to apply to Read/Write requests going to SAS-based I-T nexus. If set to zero, the timeout value is not changed.	Min: 0x1 Max: 0xFFFF Default: 0x3C Configured: 0x3C

Configuration Parameter	Description	Values
iSCSITimeOutValue	Timeout value (sec) to apply to Read/Write requests going to iSCSI-based I-T nexus. If set to zero, the timeout value is not changed.	Min: 0x1 Max: 0xFFFF Default: 0x41 Configured: 0x41

Configurable Parameters - Clustering

Table 13. Configurable Parameters - Clustering

Configuration Parameter	Description	Values
SetAPTPLForPR	A boolean value that determines whether Persistent Reservations issued by the host system will persist across a storage array power loss.	Min: 0x0 (no) Max: 0x1 (yes) Default: 0x0 Configured: 0x0

Configurable Parameters - Miscellaneous

Table 14. Configurable Parameters -Miscellaneous

Configuration Parameter	Description	Values
LoadBalancePolicy	At present, limited to specifying the default global policy to use for each MultiPath device. To override the specific MultiPath device value, change the MPIO tab found in the Device Manager <device> Properties dialog. 0x01 - Failover Only 0x03 - Round Robin with Subset 0x04 - Least Queue Depth 0x05 - Least Path Weight 0x06 - Least Blocks	Min: 0x1 Max: 0x6 Default: 0x4 Configured: 0x4
DsmMaximumStateTransitionTime	Applies only to Persistent Reservation commands. Specifies the maximum amount of time (sec) a PR request is retried during an ALUA state transition. At present, this value can be set only by directly editing the Registry.	Min: 0x0 Max: 0xFFFF Default: 0x0 Configured: 0x0
DsmDisableStatistics	Flag indicating whether per-I/O statistics are collected for use with the MPIO HEALTH_CHECK classes. At present, this value can be set only by directly editing the Registry.	Min: 0x0 (no) Max: 0x1 (yes) Default: 0x0 Configured: 0x0

Configuration Parameter	Description	Values
EventLogLevel	Formally known as 'ErrorLevel'. A bitmask controlling the category of messages which are logged. 0x00000001 - Operating System 0x00000002 - I/O Handling 0x00000004 - Failover 0x00000008 - Configuration 0x00000010 - General 0x00000020 - Troubleshooting/ Diagnostics	Min: 0x0 Max: 0x2F Default: 0x0F Configured: 0x0F

Error Handling and Event Notification

Event Logging

Event Channels

An Event Channel is a receiver (“sink”) that collects events. Some examples of event channels are the Application and System Event Logs. Information in Event Channels can be viewed through several means such as the Windows Event Viewer and **`wevtutil.exe`** command. The DSM uses a set of custom-defined channels for logging information, found under the “Applications and Services Logs" section of the Windows Event Viewer.

Custom Event View

The DSM is delivered with a custom Event Viewer filter that can combine the information from the custom-defined channels with events from the System Event Log. To use the filter, import the view from the Windows Event Viewer.

Event Messages

For the DSM, each log message is well-defined and contains one or more required **`ComponentNames`** as defined. By having a clear definition of the event log output, utilities or other applications and services can query the event logs and parse it for detailed DSM information or use it for troubleshooting purposes. The following tables list the DSM event log messages and also includes the core MPIO messages.

All MPIO-related events are logged to the System Event Log. All DSM-related events are logged to the DSM's custom Operational Event Channel.

Table 15. DSM Event Messages - Operating System Related (1000-1049)

Event Message	Event Id (Decimal)	Event Severity
Memory Allocation Error. Memory description information is in the DumpData.	1000	Informational
Queue Request Error. Additional information is in the DumpData.	1001	Informational

Table 16. DSM Event Messages - General (1050-1099)

Event Message	Event Id (Decimal)	Event Severity
<msg>. Device information is in the DumpData.	1050	Informational
<msg>. TargetPort information is in the DumpData.	1051	Informational
<msg>. TargetPortGroup information is in the DumpData.	1052	Informational
<msg>. MultipathDevice is in the DumpData.	1053	Informational
<msg>. Array information is in the DumpData.	1054	Informational
<msg>.	1055	Informational
<msg>. Device information is in the DumpData.	1056	Warning
<msg>. TargetPort information is in the DumpData.	1057	Warning
<msg>. TargetPortGroup information is in the DumpData.	1058	Warning
<msg>. MultipathDevice information is in the DumpData.	1059	Warning
<msg>. Array information is in the DumpData.	1060	Warning
<msg>.	1061	Warning
<msg>. Device information is in the DumpData.	1062	Error
<msg>. TargetPort information is in the DumpData.	1063	Error
<msg>. TargetPortGroup information is in the DumpData.	1064	Error
<msg>. MultipathDevice information is in the DumpData.	1065	Error
<msg>. Array information is in the DumpData.	1066	Error
<msg>.	1067	Error

Table 17. DSM Event Messages - I/O Handling Related (1100-1199)

Event Message	Event Id (Decimal)	Event Severity
IO Error. More information is in the DumpData.	1100	Informational
IO Request Time Exceeded. More information is in the DumpData.	1101	Informational
IO Throttle Requested to <MPIODisk_n>. More information is in the DumpData.	1102	Informational
IO Resume Requested to <MPIODisk_n>. More information is in the DumpData.	1103	Informational

Table 18. DSM Event Messages - Transfer Related (1200-1299)

Event Message	Event Id (Decimal)	Event Severity
Failover Request Issued to <MPIODisk_n>. More information is in the DumpData.	1200	Informational
Failover Request Issued Failed to <MPIODisk_n>. More information is in the DumpData.	1201	Error
Failover Request Succeeded to <MPIODisk_n>. More information is in the DumpData.	1202	Informational
Failover Request Failed to <MPIODisk_n>. More information is in the DumpData.	1203	Error

Event Message	Event Id (Decimal)	Event Severity
Failover Request Retried to <MPIODisk_n>. More information is in the DumpData.	1204	Informational
Failover Error to <MPIODisk_n>. More information is in the DumpData.	1205	Error
<MPIODisk_n> rebalanced to Preferred Target Port Group (Controller). More information is in the DumpData.	1206	Informational
Rebalance Request Failed to <MPIODisk_n>. More information is in the DumpData.	1207	Error
<MPIODisk_n> transferred due to Load Balance Policy Change. More information is in the DumpData.	1208	Informational
Transfer Due to Load Balance Policy Change Failed for <MPIODisk_n>. More information is in the DumpData.	1209	Error
Rebalance Request issued to <MPIODisk_n>. More information is in the DumpData.	1210	Informational
Rebalance Request Issued Failed to <MPIODisk_n>. Array information is in the DumpData.	1211	Error
Rebalance Request Retried to <MPIODisk_n>. More information is in the DumpData.	1212	Informational
Failover Request Issued to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1213	Informational
Failover Request Issued Failed to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1214	Error
Failover Request Failed to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1215	Error
Failover Request Retried to TargetPortGroup (Controller <n> via <MPIODisk_n>. More information is in the DumpData.	1216	Informational
Failover Setup Error for Failover to TargetPortGroup (Controller <n>). More information is in the DumpData.	1217	Error
Failover Request Succeeded to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1218	Informational
Rebalance Request issued to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1219	Informational
Rebalance Request Issued Failed to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1220	Error
Rebalance Request Retried to TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1221	Informational
Rebalance Setup Error for Rebalance to TargetPortGroup (Controller <n>). More information is in the DumpData.	1222	Error

Event Message	Event Id (Decimal)	Event Severity
<MPIODisk_n> transferred from TargetPortGroup (Controller <n>) due to Load Balance Policy Change. More information is in the DumpData.	1223	Informational
Transfer Due to Load Balance Policy Change Failed for TargetPortGroup (Controller <n>) via <MPIODisk_n>. More information is in the DumpData.	1224	Error
<MPIODisk_n> rebalance to Preferred TargetPortGroup (Controller <n>). More information is in the DumpDatta.	1225	Informational
Failure during transfer to TargetPortGroup (Controller <n>). More information is in the DumpData.	1226	Error
Transfer Setup Due to Load Balance Policy Change Failed for TargetPortGroup (Controller <n>). More information is in the DumpData.	1227	Error

Table 19. DSM Event Messages - General Configuration Related (1300-1449)

Event Message	Event Id (Decimal)	Event Severity
Configured Parameter Invalid of Out of Range. More information is in the DumpData.	1300	Informational
Configuration Initialization Error	1301	Informational
No Target Ports Found for <MPIODisk_n>. More information is in the DumpData.	1302	Error

Architecture Note:

Table 20. DSM Event Messages - Configuration Related Device Info (1450-1599)

Event Message	Event Id (Decimal)	Severity
New Device Detected. More information is in the DumpData.	1450	Informational
Device for <MPIODisk_n> Pending Removed via MPIO. More information is in the DumpData.	1451	Informational
Device for <MPIODisk_n> Removed via MPIO. More information is in the DumpData.	1452	Informational
Early Device Failure. More information is in the DumpData.	1453	Warning

Table 21. DSM Event Messages - Configuration Related Target Port (1600-1749)

Event Message	Event Id (Decimal)	Severity
New TargetPort (Path) Detected. More information is in the DumpData.	1600	Informational
TargetPort (Path) Removed via MPIO. More information is in the DumpData.	1601	Informational
TargetPort (Path) Offline Manually. More information is in the DumpData.	1602	Warning

Event Message	Event Id (Decimal)	Severity
TargetPort (Path) Online Manually. More information is found in the DumpData.	1603	Warning
TargetPort (Path) Offline (Threshold Exceeded). More information is found in the DumpData.	1604	Warning
Congestion Threshold Detected on TargetPort. More information is found in the DumpData.	1605	Warning
Not all PCD configuration parameters are set. PCD is not enabled.	1606	Warning

Table 22. DSM Event Messages - Configuration Related Target Port Group (1750-1899)

Event Message	Event Id (Decimal)	Severity
New TargetPortGroup (Controller) Detected. More information is in the DumpData.	1750	Informational
TargetPortGroup (Controller) Removed. More information is in the DumpData.	1751	Informational
TargetPortGroup (Controller) IO Timeout. More information is in the DumpData	1752	Error

Table 23. DSM Event Messages - Configuration Related Storage Array (1900-2049)

Event Message	Event Id (Decimal)	Severity
New Storage Array Detected. More information is in the DumpData.	1900	Informational
Storage Array Removed. More information is in the DumpData.	1901	Informational

Compatibility and Migration

Operating Systems Supported

The DSM is supported on Windows Server 2008 R2 and later.

Storage Interfaces Supported

The DSM supports any protocol supported by MPIO, including Fiber Channel, SAS, and iSCSI.

SAN-Boot Support

The DSM supports booting Windows from storage that is externally attached to the host.

Running the DSM in a Hyper-V Guest with Pass-Through Disks

Consider a scenario where you map storage to a Windows Server 2008 R2 parent partition. You use the **Settings** > **SCSI Controller** > **Add Hard Drive** command to attach that storage as a pass-through disk to the SCSI controller of a Hyper-V guest running Windows Server 2008. By default, some SCSI commands are filtered by Hyper-V, so the DSM Failover driver fails to run properly.

To work around this issue, you must disable SCSI command filtering. Run the following PowerShell script in the parent partition to determine if SCSI pass-through filtering is enabled or disabled:

```
# Powershell Script: Get_SCSI_Passthrough.ps1
$TargetHost=$args[0] foreach ($Child in Get-WmiObject

–Namespace root\virtualization Msvm_ComputerSystem

–Filter "ElementName='$TargetHost'") { $vmData=Get-WmiObject

 -Namespace root\virtualization -Query "Associators of {$Child}

Where ResultClass=Msvm_VirtualSystemGlobalSettingData AssocClass=Msvm_ElementSettingData"

Write-Host "Virtual Machine:" $vmData.ElementName

Write-Host "Currently Bypassing SCSI Filtering:" $vmData.AllowFullSCSICommandSet

}
```

If necessary, run the following PowerShell script in the parent partition to disable SCSI Filtering:

```
# Powershell Script: Set_SCSI_Passthrough.ps1
$TargetHost=$args[0]

$vsManagementService=gwmi MSVM_VirtualSystemManagementService

 -namespace "root\virtualization" foreach ($Child in Get-WmiObject

 -Namespace root\virtualization Msvm_ComputerSystem

-Filter "ElementName='$TargetHost'") { $vmData=Get-WmiObject

 -Namespace root\virtualization -Query "Associators of {$Child}

Where ResultClass=Msvm_VirtualSystemGlobalSettingData AssocClass=Msvm_ElementSettingData"

$vmData.AllowFullSCSICommandSet=$true

 $vsManagementService.ModifyVirtualSystem($Child,

$vmData.PSBase.GetText(1))|out-null

 }
```

Installation and Removal

Installing or Updating DSM

Perform the steps in this task to install SANtricity Storage Manager and the DSM or to upgrade from an earlier release of SANtricity Storage Manager and the DSM on a system with a Windows operating system. For a clustered system, perform these steps on each node of the system, one node at a time.

1. Open the SANtricity Storage Manager SMIA installation program, which is available from your storage vendor's website.
2. Click **Next**.
3. Accept the terms of the license agreement, and click **Next**.
4. Select **Custom**, and click **Next**.
5. Select the applications that you want to install.
6. Click the name of an application to see its description.
7. Select the check box next to an application to install it.
8. Click **Next**.

   If you have a previous version of the software installed, you receive a warning message: Existing versions of the following software already reside on this computer. If you choose to continue, the existing versions are overwritten with new versions.
9. If you receive this warning and want to update SANtricity Storage Manager, click **OK**.
10. Select whether to automatically start the Event Monitor. Click **Next**.

    Start the Event Monitor for the one I/O host on which you want to receive alert notifications. Do not start the Event Monitor for all other I/O hosts attached to the storage array or for computers that you use to manage the storage array.
11. Click **Next**.
12. If you receive a warning about anti-virus or backup software that is installed, click **Continue**.
13. Read the pre-installation summary, and click **Install**.
14. Wait for the installation to complete, and click **Done**.

Uninstalling DSM

ATTENTION To prevent loss of data, the host from which you are removing SANtricity Storage Manager and the DSM must have only one path to the storage array. Reconfigure the connections between the host and the storage array to remove any redundant connections before you uninstall SANtricity Storage Manager and the DSM failover driver.

1. From the Windows Start menu, select **Control Panel**.

   The Control Panel window appears.
2. In the Control Panel window, double-click **Add or Remove Programs**.

   The Add or Remove Programs window appears.
3. Select **SANtricity Storage Manager**.
4. Click the **Remove** button to the right of the SANtricity Storage Manager entry.

Understanding the dsmUtil Utility

The DSM solution bundles a command-line multipath utilty, named dsmUtil, to handle various management and configuration tasks. Each task is controlled through arguments on the command-line.

Reporting

The dsmUtil utility offers the following reporting options.

- **Storage Array Summary ('-a' option)** - Provides a summary of all storage arrays recognized by the DSM, and is available through the **-a** command-line option. For example, to retrieve a summary of all recognized storage arrays use the following command:

  ```
  C:\> dsmUtil -a
  ```

- **Storage Array Detail ('-a' or '-g' option)** - Provides a detailed summary of multipath devices and target ports for an array, and is available through the **-g** command-line option. The same detailed summary information is also available with an optional argument to **-a.** In either case, the array WWN is specified to obtain the detailed information as shown in the following example:

  ```
  C:\> dsmUtil -a 600a0b8000254d370000000046aaaa4c
  ```

- **Storage Array Detail Extended ('-a' or '-g' option)** - Extended information, providing further details of the configuration, is available by appending the keyword extended to the command-line for either **-a** or **-g** options. Extended information is typically used to assist in troubleshooting issues with a configuration. Extended information appears as italic but is printed as normal text output.
- **Storage Array Real-Time Status ('-S' option)** - A real-time status of the target ports between a host and array is available using the **-S** command-line option.
- **Cleanup of Status Information ('-c' option)** - Information obtained while running the **-S** option is persisted across host and array reboots. This might result in subsequent calls to the **-S** option producing erroneous results if the configuration has permanently changed. For example, a storage array is permanently removed because it is no longer needed. You can clear the persistent information using the **-c** command-line option.
- **MPIO Disk to Physical Drive Mappings ('-M' option)** - This report allows a user to cross-reference the MPIO Virtual Disk and Physical Disk instance with information from the storage array on the mapped volume. The output is similar to the smdevices utility from the SANtricity package.

Administrative and Configuration Interfaces

The dsmUtil utility offers the following administrative and configuration interface options.

- **Setting of DSM Feature Options** - Feature Options is an interface exposed by the DSM, through WMI, which can be used for several configuration parameter-related tasks.The '-o' command-line option is used to carry out these tasks. Several sub-options are available when using the '-o' option for parameter-specific purposes:
  - Parameter Listing - If the user specifies no arguments to '-o' the DSM returns a list of parameters that can be changed.
  - Change a Parameter - If the user requests a parameter value change, the DSM verifies the new parameter value, and if within range applies the value to the parameter. If the value is out of range, the DSM returns an out-of-range error condition, and dsmUtil shows an appropriate error message to the user. Note this parameter value change is in-memory only. That is, the change does not persist across a host reboot. If the user wants the change to persist, the SaveSettings option must be provided on the command-line, after all parameters have been specified.
- **Setting of MPIO-Specific Parameter** - As originally written, MPIO provided several configuration settings which were considered global to all DSMs. An enhancement was later introduced which applied some of these settings on a per-DSM basis. These settings (global and per-DSM) can be manually changed in the Registry but does not take effect until the next host reboot. They also can take effect immediately, but require that a WMI

method from a DSM-provided class is executed. For per-DSM settings, MPIO looks in the **`\\HKLM\System\CurrentControlSet\Services\<DSMName>\Parameters`** subkey. The DSM cannot invoke MPIO's WMI method to apply new per-DSM settings, therefore dsmUtil must do this. The '-P' option is used for several tasks related to MPIO's per-DSM setting.

  - Parameter Listing - An optional argument to '-P' (GetMpioParameters) is specified to retrieve the MPIO specific per-DSM settings. All of the MPIO specific settings are displayed to the user as one line in the command output.
  - Change a Parameter - If the user requests a parameter value change they provide the parameter name and new value in a 'key=value' format. Multiple parameters might be issued with a comma between each key/value statement. It appears MPIO does not do any validation of the data passed in, and the change takes effect immediately and persist across reboots.

- **Removing Device-Specific Settings** - The '-R' option is used to remove any device-specific settings for inactive devices from the Registry. Currently, the only device-specific settings that persist in the Registry are Load Balance Policy.
- **Invocation of Feature Option Actions/Methods** - Feature Options is an interface exposed by the DSM, through WMI, that also can be used to run specific actions (or methods) within the DSM. An example of an action is setting the state of a TargetPort (ie - path) to Offline. The '-o' command-line option mentioned in the Setting of Feature Options section is used to carry out these tasks. Several sub-options are available when using the '-o' option to run specific actions:
  - Action Listing - If the user specifies no arguments to '-o' the DSM returns a list of actions that can be invoked.
  - Executing An Action - Executing an action is similar to specifying a value for a configuration parameter. The user enters the name of the action, followed by a single argument to the function. The DSM runs the method and returns a success/failure status back to the utility.
- **Requesting Scan Options** - The utility can initiate several scan-related tasks. It uses the '-s' option with an optional argument that specifies the type of scan-related task to perform. Some of these are handled by the DSM while others are handled by the utility.
- **Bus Rescan** - This option causes a PnP re-enumeration to occur, and is invoked using the 'busscan' optional argument. It uses the Win32 configuration management APIs to initiate the rescan process. Communication with the DSM is not required.

Windows Multipath DSM Event Tracing and Event Logging

The DSM for Windows MPIO utilizes several methods that you can use to collect information for debugging and troubleshooting purposes. These methods are detailed in this section.

Event Tracing

The DSM for Windows MPIO uses several methods to collect information for debugging and troubleshooting purposes. These methods are detailed in this section.

About Event Tracing

Event Tracing for Windows (ETW) is an efficient kernel-level tracing facility that lets you log kernel or application-defined events to a log file. You can view the events in real time or from a log file and use the events to debug an application or to determine where performance issues are occurring in the application.

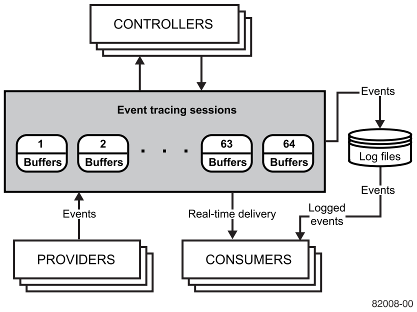
ETW lets you enable or disable event tracing dynamically, allowing you to perform detailed tracing in a production environment without requiring computer or application restarts.

The Event Tracing API is divided into three distinct components:

- Controllers, which start and stop an event tracing session and enable providers.
- Providers, which provide the events. The DSM is an example of a Provider.
- Consumers, which consume the events.

The following figure shows the event tracing model.

Figure 4. Event Tracing Overview



Controllers

Controllers are applications that define the size and location of the log file, start and stop event tracing sessions, enable providers so they can log events to the session, manage the size of the buffer pool, and obtain execution statistics for sessions. Session statistics include the number of buffers used, the number of buffers delivered, and the number of events and buffers lost.

Providers

Providers are applications that contain event tracing instrumentation. After a provider registers itself, a controller can then enable or disable event tracing in the provider. The provider defines its interpretation of being enabled or disabled. Generally, an enabled provider generates events, while a disabled provider does not. This lets you add event tracing to your application without requiring that it generate events all the time. Although the ETW model separates the controller and provider into separate applications, an application can include both components.

There are two types of providers: the classic provider and the manifest-based provider. The DSM is a classic provider and the tracing events it generates are from the 'TracePrint' API.

Consumers

Consumers are applications that select one or more event tracing sessions as a source of events. A consumer can request events from multiple event tracing sessions simultaneously; the system delivers the events in chronological order. Consumers can receive events stored in log files, or from sessions that deliver events in real time. When processing events, a consumer can specify start and end times, and only events that occur in the specified time frame will be delivered.

What You Need to Know About Event Tracing

- Event Tracing uses Non-Paged Pool kernel memory to hold the unflushed events. When configuring trace buffer sizes, try to minimize the buffers potentially used.
- If large trace buffer sizes have been requested at boot, you might experience a delay in boot-time as referenced in this knowledge base article: http://support.microsoft.com/kb/2251488.
- If events are being added to the trace buffer faster than can be flushed then you can experience missed events. The logman utility indicates how many events are missed. If you experience this behavior, either increase your trace buffer size or (if flushing to a device) find a device that can handle faster flush rates.

Collecting Trace Events from a Target Machine

There are several utilities and tools that can be used to collect Trace Events. These tools and utilities typically establish a new trace session, along with specifying what flags and level of tracing to capture. When capturing is complete, the trace session is stopped and the capture buffers flushed of any cached information.

Control Files

Several tools and utilities require knowing the GUID of the provider as well as trace flags and level. If you want only to collect information for a single provider, you can provide the GUID and trace settings through one or more command-line arguments. To capture from multiple sources, use Control Files. The Control File format is typically:

```
{GUID} [Flags Level]
```

For example:

```
C:>type mppdsm.ctl
```

```
{706a8802-097d-43C5-ad89-8863e84774c6} 0x0000FFFF 0xF
```

Logman

The Logman tool manages and schedules performance counter and event trace log collections on local and remote systems, and is provided in-box with each OS installation.There is no explicit requirement for the DSM Trace Provider to be registered before you can use Logman to capture trace events, although for end-user convenience the DSM should be registered during installation.

Viewing a List of Available Providers

To view a list of available providers:

```
C:>logman query providers
```

By default the DSM does not show up in this list unless it has previously been registered.

Establishing a New Trace Session

To establish a new trace session:

```
C:>logman create trace <session_name> -ets -nb 16 256 -bs 64 -o <logfile> -pf <control_file>
```

Where:

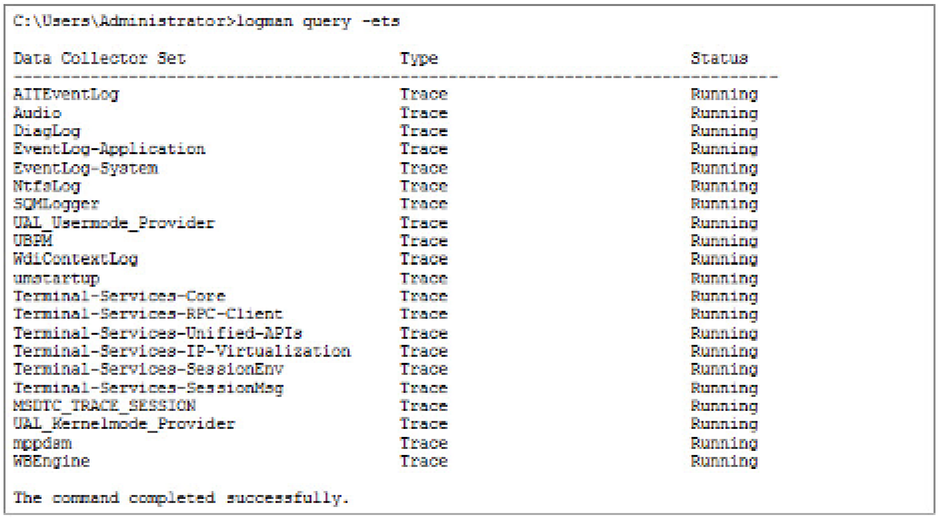
- <**session_name**>: Name of the trace session (ex. "mppdsm")

- <**`control_file`**>: Trace control file.

Determine Status of Trace Sessions

To determine whether a trace session is running, using the 'query' option. In this example an 'mppdsm' trace session has been created and shown as running:

Figure 5. Command to determine if a trace session is running

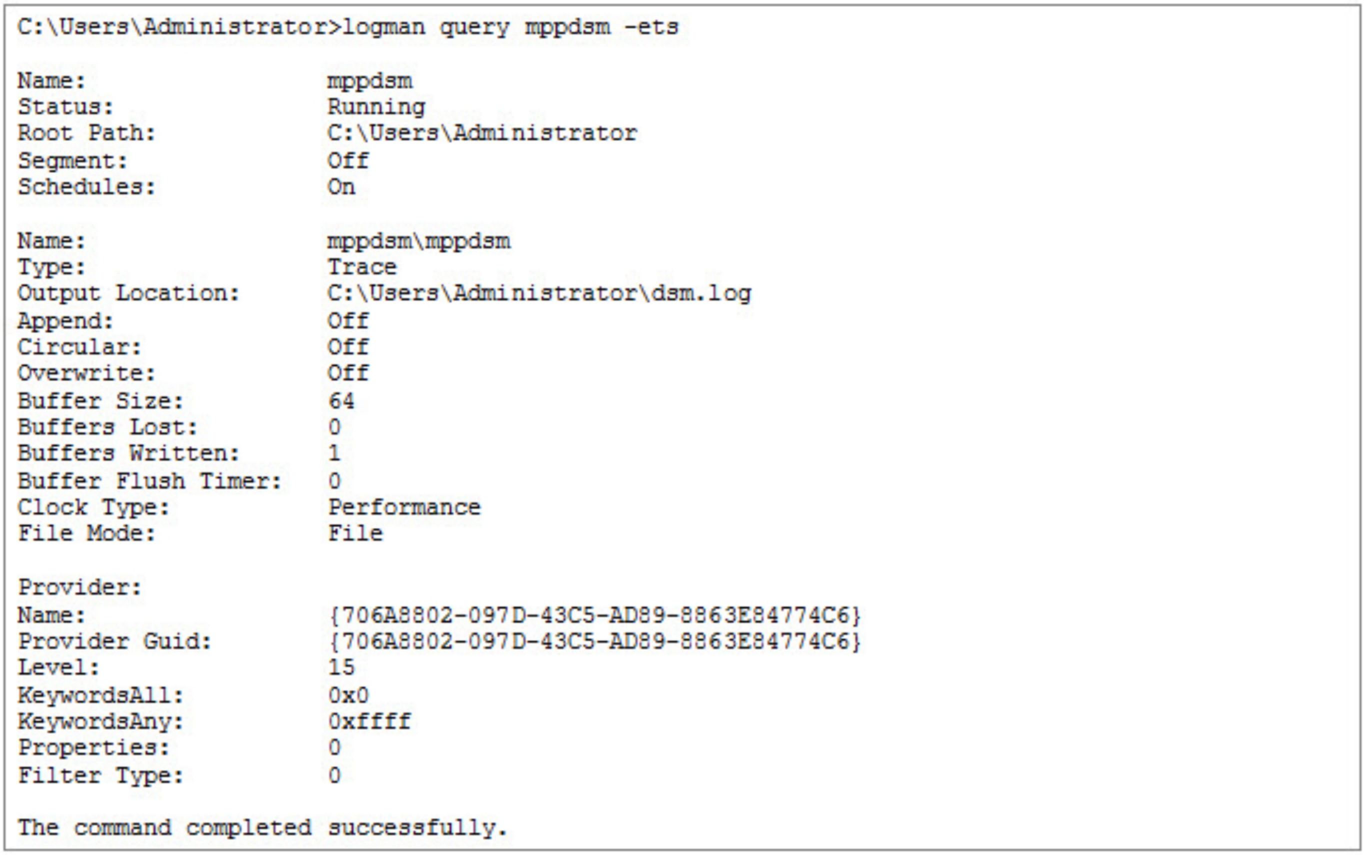


```
C:\Users\Administrator>logman query -ets

Data Collector Set                      Type                          Status
-------------------------------------------------------------------------------
AITEventLog                             Trace                         Running
Audio                                   Trace                         Running
DiagLog                                 Trace                         Running
EventLog-Application                    Trace                         Running
EventLog-System                         Trace                         Running
NtfsLog                                 Trace                         Running
SQMLogger                               Trace                         Running
UAL_Usermode_Provider                   Trace                         Running
UBPM                                    Trace                         Running
WdiContextLog                           Trace                         Running
umstartup                               Trace                         Running
Terminal-Services-Core                  Trace                         Running
Terminal-Services-RPC-Client            Trace                         Running
Terminal-Services-Unified-APIs          Trace                         Running
Terminal-Services-IP-Virtualization     Trace                         Running
Terminal-Services-SessionEnv            Trace                         Running
Terminal-Services-SessionMsg            Trace                         Running
MSDTC_TRACE_SESSION                     Trace                         Running
UAL_Kernelmode_Provider                 Trace                         Running
mppdsm                                  Trace                         Running
WBEngine                                Trace                         Running

The command completed successfully.
```

82008-04

The following command can be used to get more detailed information about the trace session. In this example, the 'mppdsm' session is detailed:

Figure 6. Command to retrieve more information about the trace session



```
C:\Users\Administrator>logman query mppdsm -ets

Name:                   mppdsm
Status:                 Running
Root Path:              C:\Users\Administrator
Segment:                Off
Schedules:              On

Name:                   mppdsm\mppdsm
Type:                   Trace
Output Location:        C:\Users\Administrator\dsm.log
Append:                 Off
Circular:               Off
Overwrite:              Off
Buffer Size:            64
Buffers Lost:           0
Buffers Written:        1
Buffer Flush Timer:     0
Clock Type:             Performance
File Mode:              File

Provider:
Name:                   {706A8802-097D-43C5-AD89-8863E84774C6}
Provider Guid:          {706A8802-097D-43C5-AD89-8863E84774C6}
Level:                  15
KeywordsAll:            0x0
KeywordsAny:            0xffff
Properties:             0
Filter Type:            0

The command completed successfully.
```

82008-05

Stopping a Trace Session

To stop a tracing session:

```
C:\Users\Administrator>logman stop <session_name> -ets

The command completed successfully.
```

Deleting a Trace Session

To delete a tracing session:

```
C:\Users\Administrator>logman delete <session_name>

The command completed successfully.
```

Enabling a Boot-Time Trace Session

Enabling boot-time tracing is done by appending "autosession" to the session name:

```
logman create trace "autosession\<session_name>"

-o <logfile> -pf <control_file>
```

For example:

```
C:\Users\Administrator>logman create trace "autosession\mppdsm"

-o mppdsmtrace.etl -pf mppdsm.ctl

The command completed successfully.
```

Boot-Time sessions can be stopped and deleted just like any other session.

NOTE You need to register the DSM as a provider with WMI or boot-time logging does not occur.

Disabling a Boot-Time Trace Session

To disable a boot-time trace session:

```
C:\Users\Administrator\logman delete "autosession\mppdsm"

The command completed successfully.
```

Viewing Trace Events

Trace events captured to a log file are in a binary format that is not human-readable, but can be decoded properly by Technical Support. Submit any captured logs to your Technical Support Representative.

Event Logging

Windows Event Logging provides applications and the operating system a way to record important software and hardware events. The event logging service can record events from various sources and store them in a single collection called an Event Log. The Event Viewer, found in Windows, enables users to view these logs. Version 1.x of the DSM recorded events in the legacy system log.

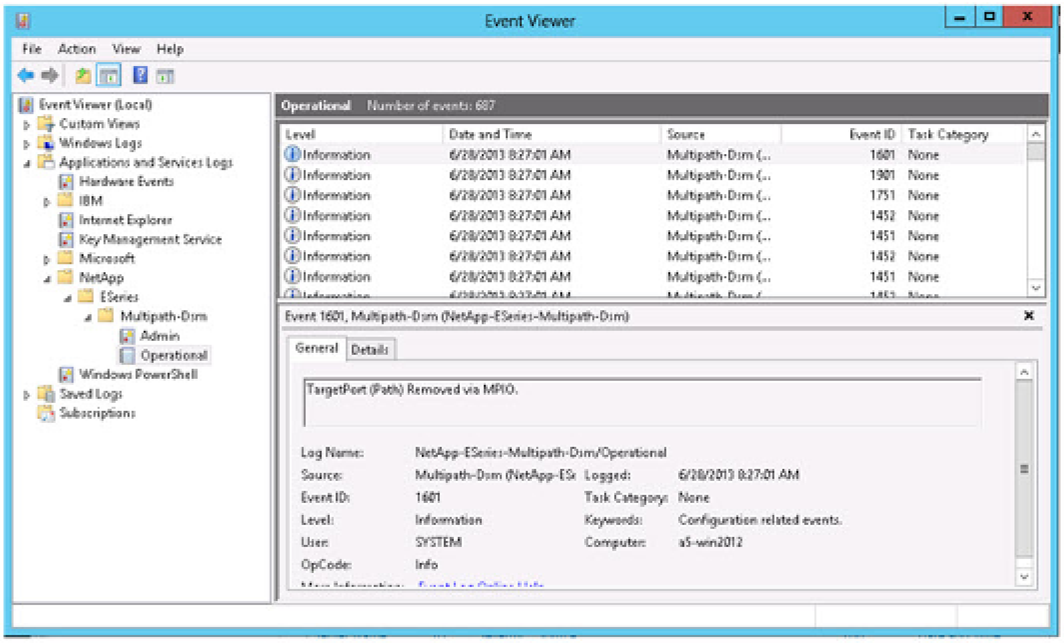
Windows Server 2008 introduced a redesign of the event logging structure that unified the Event Tracing for Windows (ETW) and Event Log APIs. It provides a more robust and powerful mechanism for logging events. Version 2.x of the DSM uses this new approach.

As with Event Tracing, the DSM is considered a provider of Event Log events. Event Log events can be written to the legacy system log, or to new event channels. These event channels are similar in concept to the legacy system log but allow the DSM to record more detailed information about each event generated. In addition, it allows the DSM to record the information into a dedicated log where it won't overwrite or obscure events from other components in the system. Event channels also can support the ability to write events at a higher throughput rate.

Event Channels

Event channels are viewed using the same Event Viewer application that you use to view the legacy system logs. Currently, the only channel used is the Operational channel. Events logged into the Admin and Operational channels are stored in the same **.EVTX** format used by other Windows logs. The following figure shows an example of the event channels.

Figure 7. Event Channels



Level	Date and Time	Source	Event ID	Task Category
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1601	None
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1901	None
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1751	None
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1452	None
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1451	None
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1452	None
Information	6/28/2013 8:27:01 AM	Multipath-Dsm (...	1451	None

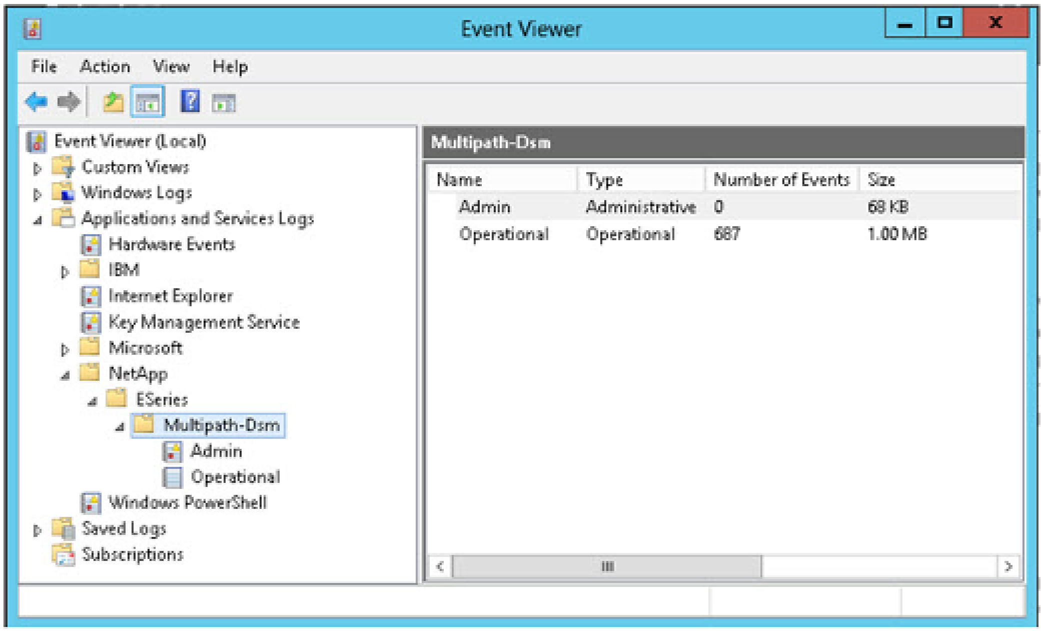
TargetPort (Path) Removed via MPIO.

Log Name: NetApp-ESeries-Multipath-Dsm/Operational
Source: Multipath-Dsm (NetApp-ESe
Logged: 6/28/2013 8:27:01 AM
Event ID: 1601
Task Category: None
Level: Information
Keywords: Configuration related events.
User: SYSTEM
Computer: a5-win2012
OpCode: Info

82008-02

When you select the Operational channel, a tri-pane window appears that shows several rows of events and details of the currently selected event as shown in the following figure. You can select the Details tab to view the raw XML data that makes up the event.

Figure 8. Event Channel Detail



Name	Type	Number of Events	Size
Admin	Administrative	0	68 KB
Operational	Operational	687	1.00 MB

82008-03

Loading the Custom Event View

You can use the custom view to combine the DSM and system log information into a single view.

1. In the Event Viewer application, right-click **Custom Views** > **Import Custom View**.
2. Go to the directory where the DSM installation is installed and look in the 'drivers' directory for a file named **`CombinedDsmEventChannelView.xml`**.
3. Click **OK** to accept the location of the custom view.

   A new Custom View named **`CombinedDsmEventChannelView`** will appear as an option. Select the new custom view to show output from both logs.

Event Decoding

Version 2.x of the DSM provides an internally-consistent way of storing information about an object, such as a disk device or controller, which can be provided as part of each record written to an event channel. The component information is a raw stream of bytes that is decoded and merged with the other data to present a complete description of each event record.

1. When the DSM solution is built, the source code is scanned by a script which generates several XML definition files describing details of each Event and the associated base components. These XML definition files are shipped with the solution.

2. Events that need to be decoded are saved to an **`.EVTX`** file, or can be decoded directly on a Host if there is access to the required Event channels.
3. A PowerShell script and **`cmdlet`** uses the XML and Event Logs to generate a CSV-formatted document containing the decoded events. This document can be imported to applications such as Excel for viewing.

Files Used in the Decode Process

The 'decoder' directory contains all the files used to decode the event logs.

- **`'DecodeEvents.bat`** - This batch file invokes a new powershell session to execute the decoding process. The decoding process will utilize the XML files described below.
- **`BaseComponents.xml`**- This XML file provides details on each base component and should not be modified as any change can cause a failure in properly decoding events.
- **`EventComponents.xml`**- This XML file provides details for each event generated by the DSM and the base component data reported. It should not be modified as any change can cause a failure in properly decoding events.
- **`LogsToDecode.xml`** - This XML file defines the source(s) of the event log data. For convenience the decoding process will not only attempt to decode messages from the DSM, but also messages reported by Microsoft MPIO. This file can be modified as needed to define the location of event log data to decode.
- **`DsmEventDecoder.psm1`** - The powershell module, which queries the event logs for information, calls the **`FormatDsmEventLog cmdlet`** to parse and decode the event information.

Decoded Output

The information decoded into a CSV format consists of several sections as described below.

1. The first section describes the input arguments to the powershell decoder script.
2. The second section is a detailed dump of the BaseComponent and EventComponent XML files. You can use this section to manually decode the event data if the automated process runs into an error with the event data. This section is also useful if only the decoded results are provided to Technical Support rather than the original *.EVTX files.
3. The last section is the actual decoded events. Note that the entire event log is decoded, not just the event specific information. Furthermore, an attempt to decode the Microsoft MPIO-generated events is provided for convenience.

Limitations

The following items list the limitations for the decoding process.

- If a large number of records are present the decoding process may take some time.
- CSV format is currently the only supported output format.

Failover Drivers for the Linux Operating System

The following failover drivers are supported with the Linux operating system.

- Redundant Dual Active Controller (RDAC) is the failover driver for the Linux operating system that is included with the SANtricity® Storage Manager. Storage arrays using this failover driver should use the Linux (MPP/RDAC) host type.
- Device Mapper Multipath (DMMP) failover, which uses the Device Mapper generic framework for mapping one block device onto another. Device mapper is used for LVM, multipathing, and more.
- **`The scsi_dh_rdac`** plug-in with DMMP is a multipathing driver that is used to communicate with storage arrays. It provides an ALUA solution when used with CFW version 7.83 and later. Storage arrays using this failover driver should use the Linux (DM-MP) host type.

MPP/RDAC Linux Host Type

MPP/RDAC is the Linux host type supported for the RDAC failover driver for the Linux operating system.

The RDAC failover driver is not the recommended failover driver for the Linux operating system. The RDAC failover driver will be deprecated in a future release. Refer to the instructions in the Migrating to the Linux DMMP Driver topic to learn how to migrate to a failover driver that will continue to be supported in future releases.

Features of the RDAC Failover Driver Provided with the SANtricity Storage Manager

Redundant Dual Active Controller (RDAC), or MPP-RDAC, is the failover driver for the Linux OS that is included in SANtricity Storage Manager. The RDAC failover driver includes these features:

- On-the-fly path validation.
- Cluster support.
- Automatic detection of path failure. The RDAC failover driver automatically routes I/O to another path in the same controller or to an alternate controller, in case all paths to a particular controller fail.
- Retry handling is improved, because the RDAC failover driver can better understand vendor-specific statuses returned from the controller through sense key/ASC/ASCQ.
- Automatic rebalance is handled. When the failed controller obtains Optimal status, storage array rebalance is performed automatically without user intervention.
- Load-balancing policies including round robin subset and least queue depth.

RDAC Load Balancing Policies

Load balancing is the redistribution of read/write requests to maximize throughput between the server and the storage array. Load balancing is very important in high workload settings or other settings where consistent service levels are critical. The multi-path driver transparently balances I/O workload without administrator intervention. Without multipath software, a server sending I/O requests down several paths might operate with very heavy workloads on some paths, while other paths are not used efficiently.

The multi-path driver determines which paths to a device are in an active state and can be used for load balancing. Multiple options for setting the load-balancing policies allow you to optimize I/O performance when mixed host interfaces are configured. Load balancing is performed on multiple paths to the same controller but not across both controllers.

The load-balancing policies that you can select for the RDAC multi-path driver include the following.

- **Round Robin Subset** - The round robin with subset I/O load-balancing policy routes I/O requests, in rotation, to each available data path to the controller that owns the volumes. This policy treats all paths to the controller that owns the volume equally for I/O activity. Paths to the secondary controller are ignored until ownership changes. The basic assumption for the round robin with subset I/O policy is that the data paths are equal. With mixed host support, the data paths might have different bandwidths or different data transfer speeds.
- **Least Queue Depth** - The least queue depth policy is also known as the least I/Os policy or the least requests policy. This policy routes the next I/O request to the data path on the controller that owns the volume that has the least outstanding I/O requests queued. For this policy, an I/O request is simply a command in the queue. The type of command or the number of blocks that are associated with the command is not considered. The least queue depth policy treats large block requests and small block requests equally. The data path selected is one of the paths in the path group of the controller that owns the volume.

Prerequisites for Installing RDAC on the Linux OS

Before installing RDAC on the Linux OS, make sure that your storage array meets these conditions:

- Make sure that the host system on which you want to install the RDAC driver has supported HBAs.
- Refer to the appropriate installation guide for your controller tray or base system for any configuration settings that you need to make.
- Although the system can have Fibre Channel HBAs from multiple vendors or multiple models of Fibre Channel HBAs from the same vendor, you can connect only the same model of Fibre Channel HBAs to each storage array.
- Make sure that the low-level HBA driver has been correctly built and installed before RDAC driver installation.
- The standard HBA driver must be loaded before you install the RDAC driver. The HBA driver has to be a non-failover driver.
- For LSI HBAs, the port driver is named `mptbase`, and the host driver is named `mptscsi` or `mptscsih`, although the name depends on the driver version. The Fibre Channel driver is named `mptfc`, the SAS driver is named `mptsas`, and the SAS2 driver is named `mpt2sas`.
- For QLogic HBAs, the base driver is named `qla2xxx`, and host driver is named `qla2300`. The 4-GB HBA driver is named `qla2400`.
- For IBM Emulex HBAs, the base driver is named `lpfcdd` or `lpfc`, although the name depends on the driver version.
- For Emulex HBAs, the base driver is named `lpfcdd` or `lpfc`, although the name depends on the driver version.
- Make sure that the kernel source tree for the kernel version to be built against is already installed. You must install the kernel source rpm on the target system for the SUSE SLES OS. You are not required to install the kernel source for the Red Hat OS.
- Make sure that the necessary kernel packages are installed: `source rpm` for the SUSE OS and `kernel headers/kernel devel` for the Red Hat Enterprise Linux OS.

In SUSE OSs, you must include these items for the HBAs mentioned as follows:

- For LSI HBAs, INITRD_MODULES includes `mptbase` and `mptscsi` (or `mptscsih`) in the `/etc/sysconfig/kernel` file. The Fibre Channel driver is named `mptfc`, the SAS driver is named `mptsas`, and the SAS2 driver is named `mpt2sas`.
- For QLogic HBAs, INITRD_MODULES includes a `qla2xxx` driver and a `qla2300` driver in the `/etc/sysconfig/kernel` file.
- For IBM Emulex HBAs, INITRD_MODULES includes an `lpfcdd` driver or an `lpfc` driver in the `/etc/sysconfig/kernel` file.
- For Emulex HBAs, INITRD_MODULES includes an `lpfcdd` driver or an `lpfc` driver in the `/etc/sysconfig/kernel` file.

Installing SANtricity Storage Manager and RDAC on the Linux OS

IMPORTANT SANtricity Storage Manager requires that the different Linux OS kernels have separate installation packages. Make sure that you are using the correct installation package for your particular Linux OS kernel.

1. Open the SANtricity Storage Manager SMIA installation program, which is available from your storage vendor's website.

   The SANtricity Storage Manager installation window appears.
2. Click **Next**.
3. Accept the terms of the license agreement, and click **Next**.
4. Select one of the installation packages:
   - **Typical** – Select this option to install all of the available host software.
   - **Management Station** – Select this option to install software to configure, manage, and monitor a storage array. This option does not include RDAC. This option installs only the client software.
   - **Host** – Select this option to install the storage array server software.
   - **Custom** – Select this option to customize the features to be installed.
5. Click **Next**.

   NOTE For this procedure, **Typical** is selected. If the **Host** installation option is selected, the Agent, the Utilities, and the RDAC driver are installed.

   You might receive a warning after you click **Next**. The warning states:

   ```
   Existing versions of the following software already reside on
   this computer ... If you choose to continue, the existing
   versions will be overwritten with new versions ....
   ```

   If you receive this warning and want to update the SANtricity Storage Manager Version, click **OK**.
6. Click **Install**.

   A warning appears after you click **Install**. The warning tells you that the RDAC driver is not automatically installed. You must manually install the RDAC driver.

   The RDAC source code is copied to the specified directory in the warning message. Go to that directory, and perform the steps in Installing RDAC Manually on the Linux OS.
7. Click **Done**.

Installing RDAC Manually on the Linux OS

1. To unzip the RDAC `tar.gz` file and enter the RDAC tar file, type this command, and press **Enter**:

   ```
   tar -zxvf <filename>
   ```

2. Go to the Linux RDAC directory.
3. Type this command, and press **Enter**.

   ```
   make uninstall
   ```

4. To remove the old driver modules in that directory, type this command, and press **Enter**:

   ```
   make clean
   ```

5. To compile all driver modules and utilities in a multiple CPU server (SMP kernel), type this command, and press **Enter**:

   ```
   make
   ```

6. Type this command, and press **Enter**:

   ```
   make install
   ```

   These actions result from running this command:

   - The driver modules are copied to the kernel module tree.
   - The new RAMdisk image (`mpp-`uname -r`.img`) is built, which includes the RDAC driver modules and all driver modules that are needed at boot.

7. Follow the instructions shown at the end of the build process to add a new boot menu option that uses `/boot/mpp-`uname -r`.img` as the initial RAMdisk image.

Making Sure that RDAC Is Installed Correctly on the Linux OS

1. Restart the system by using the new boot menu option.
2. Make sure that these drivers were loaded after restart by running the lsmod command:
   - `scsi_mod`
   - `sd_mod`
   - `sg`
   - `mppUpper`
   - The physical HBA driver module (lpfc, mptsas, mpt2sas, qla2xxxx)
   - `mppVhba`
3. To verify that the MPP driver has discovered the available physical volumes and created virtual volumes for them, type this command, and press **Enter**:

   ```
   /opt/mpp/lsvdev
   ```

   You can now send I/O to the volumes.

4. If you make any changes to the RDAC configuration file (`/etc/mpp.conf`) or the persistent binding file (`/var/mpp/devicemapping`), run the **`mppUpdate`** command to rebuild the RAMdisk image to include the new file. In this way, the new configuration file (or persistent binding file) can be used on the next system restart.

5. To dynamically reload the driver stack (`mppUpper`, physical HBA driver modules, `mppVhba`) without restarting the system, perform these steps:
    a. Remove all of the configured scsi devices from the system.
    b. To unload the `mppVhba` driver, type this command, and press **Enter**:
    ```
    modprobe -r
    ```
    c. To unload the physical HBA driver, type this command, and press **Enter**:
    ```
    modprobe -r "physical hba driver modules"
    ```
    d. To unload the `mppUpper` driver, type this command, and press **Enter**:
    ```
    modprobe -r
    ```
    e. To reload the `mppUpper` driver, type this command, and press **Enter**:
    ```
      modprobe mppUpper
    ```
    f. To reload the physical HBA driver, type this command, and press **Enter**:
    ```
      modprobe "physical hba driver modules"
    ```
    g. To reload the `mppVhba` driver, type this command, and press **Enter**:
    ```
      modprobe mppVhba
    ```
6. Restart the system whenever there is an occasion to unload the driver stack.
7. Use a utility, such as **`devlabel`**, to create user-defined device names that can map devices based on a unique identifier, called a UUID.
8. Use the **udev** command for persistent device names. The **udev** command dynamically generates device name links in the `/dev/disk` directory based on path, ID or UUID.

    ```
    linux-kbx5:/dev/disk # ls /dev/disk by-id  by-path  by-uuid
    ```

    For example, the `/dev/disk/by-id` directory links volumes that are identified by WWIDs of the volumes to actual disk device nodes.

    ```
    lrwxrwxrwx 1 root root 10 Feb 23 12:15
    scsi-3600a0b80000c2df9000003b141417799 -> ../../sdda

    lrwxrwxrwx 1 root root  9 Feb 23 12:15
    scsi-3600a0b80000f27030000000d416b94fd -> ../../sdc

    lrwxrwxrwx 1 root root  9 Feb 23 12:15
    scsi-3600a0b80000f270300000015416b958f -> ../../sdg
    ```

Configuring Failover Drivers for the Linux OS

The Windows OS and the Linux OS share the same set of tunable parameters to enforce the same I/O behaviors.

Table 24. Configuration Settings for Failover Drivers for the Linux OS

Parameter Name	Default Value	Description
`ImmediateVirtLunCreate`	**`0`**	This parameter determines whether to create the virtual LUN immediately if the owning physical path is not yet discovered. This parameter can take the following values: ■ **`0`** – Do not create the virtual LUN immediately if the owning physical path is not yet discovered. ■ **`1`** – Create the virtual LUN immediately if the owning physical path is not yet discovered.
`BusResetTimeout`		The time, in seconds, for the RDAC driver to delay before retrying an I/O operation if the DID_RESET status is received from the physical HBA. A typical setting is **`150`**.
`AllowHBAsgDevs`	**`0`**	This parameter determines whether to create individual SCSI generic (SG) devices for each I:T:L for the end LUN through the physical HBA. This parameter can take the following values: ■ **`0`** – Do not allow creation of SG devices for each I:T:L through the physical HBA. ■ **`1`** – Allow creation of SG devices for each I:T:L through the physical HBA.

mppUtil Utility

The mppUtil utility is a general-purpose command-line driven utility that works only with MPP-based RDAC solutions. The utility instructs RDAC to perform various maintenance tasks but also serves as a troubleshooting tool when necessary.

To use the mppUtil utility, type this command, and press **Enter**:

```
mppUtil [-a target_name] [-c wwn_file_name] [-d debug_level]
[-e error_level] [-g virtual_target_id] [-I host_num]
[-o feature_action_name[=value][, SaveSettings]]
[-s "failback" | "avt" | "busscan" | "forcerebalance"] [-S] [-U]
[-V] [-w target_wwn,controller_index]
```

NOTE The quotation marks must surround the parameters.

The mppUtil utility is a cross-platform tool. Some parameters might not have a meaning in a particular OS environment. A description of each parameter follows.

Table 25. mppUtil Parameters

Parameter	Description
`-a` *`target_name`*	Shows the RDAC driver's internal information for the specified virtual *`target_name`* (storage array name). If a *`target_name`* value is not included, the `-a` parameter shows information about all of the storage arrays that are currently detected by this host.
`-c` *`wwn_file_name`*	Clears the WWN file entries. This file is located at `/var/mpp` with the extension `.wwn`.
`-d` *`debug_level`*	Sets the current debug reporting level. This option works only if the RDAC driver has been compiled with debugging enabled. Debug reporting is comprised of two segments. The first segment refers to a specific area of functionality, and the second segment refers to the level of reporting within that area. The *`debug_level`* is one of these hexadecimal numbers: ■ **`0x20000000`** – Shows messages from the RDAC driver's init() routine. ■ **`0x10000000`** – Shows messages from the RDAC driver's attach() routine. ■ **`0x08000000`** – Shows messages from the RDAC driver's ioctl() routine. ■ **`0x04000000`** – Shows messages from the RDAC driver's open() routine. ■ **`0x02000000`** – Shows messages from the RDAC driver's read() routine. ■ **`0x01000000`** – Shows messages related to HBA commands. ■ **`0x00800000`** – Shows messages related to aborted commands. ■ **`0x00400000`** – Shows messages related to panic dumps. ■ **`0x00200000`** – Shows messages related to synchronous I/O activity. ■ **`0x00000001`** – Debug level 1. ■ **`0x00000002`** – Debug level 2. ■ **`0x00000004`** – Debug level 3. ■ **`0x00000008`** – Debug level 4. These options can be combined with the logical AND operator to provide multiple areas and levels of reporting as needed. For use by Technical Support Representatives only.

Parameter	Description
`-e` *`error_level`*	Sets the current error reporting level to *`error_level`*, which can have one of these values: ■ **`0`** – Show all errors. ■ **`1`** – Show path failover, controller failover, retryable, fatal, and recovered errors. ■ **`2`** – Show path failover, controller failover, retryable, and fatal errors. ■ **`3`** – Show path failover, controller failover, and fatal errors. This is the default setting. ■ **`4`** – Show controller failover and fatal errors. ■ **`5`** – Show fatal errors. For use by Technical Support Representatives only.
`-g` *`virtual_target_id`*	Shows the RDAC driver's internal information for the specified *`virtual_target_id`*.
`-I` *`host_num`*	Prints the maximum number of targets that can be handled by that host. Here, host refers to the HBA drivers on the system and includes the RDAC driver. The host number of the HBA driver is given as an argument. The host numbers assigned by the Linux middle layer start from 0. If two ports are on the HBA card, host numbers 0 and 1 would be taken up by the low-level HBA driver, and the RDAC driver would be at host number 2. Use `/proc/scsi` to determine the host number.

`-o` *`feature_action_name`*`[=`*`value`*`]`
`[, SaveSettings]`
	Troubleshoots a feature or changes a configuration setting. Without the `SaveSettings` keyword, the changes affect only the in-memory state of the variable. The `SaveSettings` keyword changes both the in-memory state and the persistent state. You must run **`mppUpdate`** to reflect these changes in the inird image before rebooting the server. Some example commands are:
	■ **`mppUtil -o`** – Shows all the available feature action names.
	■ **`mppUtil -o ErrorLevel=0x2`** – Sets the `ErrorLevel` parameter to **`0x2`** (affects only the in-memory state).

`-s ["failback" | "avt" |`
`"busscan" | "forcerebalance"]`
	Manually initiates one of the RDAC driver's scan tasks.
	■ A "failback" scan causes the RDAC driver to reattempt communications with any failed controllers.
	■ An "avt" scan causes the RDAC driver to check whether AVT has been enabled or disabled for an entire storage array.
	■ A "busscan" scan causes the RDAC driver to go through its unconfigured devices list to see if any of them have become configured.
	■ A "forcerebalance" scan causes the RDAC driver to move storage array volumes to their preferred controller and ignore the value of the `DisableLunRebalance` configuration parameter of the RDAC driver.

Parameter	Description
`-S`	Reports the Up state or the Down state of the controllers and paths for each LUN in real time.
`-U`	Refreshes the Universal Transport Mechanism (UTM) LUN information in MPP driver internal data structure for all the storage arrays that have already been discovered.
`-V`	Prints the version of the RDAC driver currently running on the system.
`-w` *`target_wwn,controller_index`*	For use by Technical Support Representatives only.

Frequently Asked Questions about MPP/RDAC

Table 26. Frequently Asked Questions about Linux Failover Drivers

Question	Answer
How do I get logs from RDAC in the Linux OS?	Use the **`mppSupport`** command to obtain several logs related to RDAC. The **`mppSupport`** command is found in the `/opt/ mpp/mppSupport` directory. The command creates a file named `mppSupportdata_`*`hostname_RDAC version_datetime`*`.tar.gz` in the `/tmp` directory.
How does persistent naming work?	The Linux OS SCSI device names can change when the host system restarts. Use a utility, such as devlabel, to create user-defined device names that will map devices based on a unique identifier. The udev method is the preferred method.
What must I do after applying a kernel update?	After you apply the kernel update and start the new kernel, perform these steps to build the RDAC Initial Ram Disk image (initrd image) for the new kernel: 1. Change the directory to the Linux RDAC source code directory. 2. Type `make uninstall`, and press **Enter**. 3. Reinstall RDAC. Go to Installing RDAC Manually on the Linux OS.
What is the Initial Ram Disk Image (initrd image), and how do I create a new initrd image?	The initrd image is automatically created when the driver is installed by using the **`make install`** command. The boot loader configuration file must have an entry for this newly created image. The initrd image is located in the boot partition. The file is named `mpp‘-uname -r’.img`. For a driver update, if the system already has a previous entry for RDAC, the system administrator must modify the existing RDAC entry in the boot loader configuration file. In most of the cases, no change is required if the kernel version is the same. To create a new initrd image, type `mppUpdate`, and press **Enter**. The old image file is overwritten with the new image file. For the SUSE OS, if third-party drivers need to be added to the initrd image, change the `/etc/sysconfig/ kernel` file with the third-party driver entries. Run the **`mppUpdate`** command again to create a new initrd image.

Question	Answer
How do I remove unmapped or disconnected devices from the existing host?	Run **`hot_add -d`** to remove all unmapped or disconnected devices.
What if I remap a LUN from the storage array?	Run **`hot_add -u`** to update the host with the changed LUN mapping.
What if I change the size of the LUN on the storage array?	Run **`hot_add -c`** to change the size of the LUN on the host.

How do I know what storage arrays MPP has discovered?	To make sure that the RDAC driver has found the available storage arrays and created virtual storage arrays for them, type these commands, and press **Enter** after each command.
	`ls -lR /proc/mpp`
	`mppUtil -a`
	`/opt/mpp/lsvdev`
	To show all attached and discovered volumes, type `cat /proc/scsi/scsi`, and press **Enter**.

What should I do if I receive this message?
`Warning: Changing the storage array name can cause host applications to lose access to the storage array if the host is running certain path failover drivers.`
`If any of your hosts are running path failover drivers, please update the storage array name in your path failover driver's configuration file before rebooting the host machine to insure uninterrupted access to the storage array. Refer to your path failover driver documentation for more details.`
	The path failover drivers that cause this warning are the RDAC drivers on both the Linux OS and the Windows OS.
	The storage array user label is used for storage array-to-virtual target ID binding in the RDAC driver. For the Linux OS, change this file to add the storage array user label and its virtual target ID.
	`.~ # more /var/mpp/devicemapping`

Device Mapper Multipath (DMMP) for the Linux Operating System

Device Mapper Multipath (DMMP) is a generic framework for block devices provided by the Linux operating system. It supports concatenation, striping, snapshots (legacy), mirroring, and multipathing. The multipath function is provided by the combination of the kernel modules and user space tools.

Device Mapper Features

- Provides a single block device node for a multipathed logical unit
- Ensures that I/O is re-routed to available paths during a path failure
- Ensures that the failed paths are revalidated as soon as possible
- Configures the multipaths to maximize performance
- Reconfigures the multipaths automatically when events occur
- Provides DMMP features support to newly added logical unit
- Provides device name persistence for DMMP devices under `/dev/mapper/`
- Configures multipaths automatically at an early stage of rebooting to permit the OS to install and reboot on a multipathed logical unit

DMMP Load Balancing Policies

The load-balancing policies that you can select for the DMMP multi-path driver include the following.

- Round robin 0: Loops through every path in the path group, sending the same amount of I/O to each.
- Queue length 0: Sends the next group of I/O down the path with the least amount of outstanding I/O.
- Service time 0: Selects the path for the next group of I/O based on the amount of outstanding I/O to the path and its relative throughput.

Known Limitations and Issues of the Device Mapper

- In certain error conditions, with `no_path_retry` or `queue_if_no_path` feature set, applications might hang forever. To overcome these conditions, you must enter the following command to all the affected multipath devices: **`dmsetup message device 0 "fail_if_no_path"`**, where **`device`** is the multipath device name (for example, mpath2; do not specify the path).
- Normally, the scsi_dh_rdac module is not included in initrd image. When this module is not included, boot-up might be very slow with large configurations and the `syslog` file might contain a large number of buffer I/O errors. You should include scsi_dh_rdac in the initrd to avoid these problems. The installation procedures for each operating system describe how to include the driver in initrd image. Refer to the appropriate procedure in Device Mapper Operating Systems Support.
- If the storage vendor and model are not included in scsi_dh_rdac device handler, device discovery might be slower, and the `syslog` file might get populated with buffer I/O error messages.
- Use of the DMMP and RDAC failover solutions together on the same host is not supported. Use only one solution at a time.
- DMMP is not capable of detecting changes by itself when the user changes LUN mappings on the target.
- DMMP is not capable of detecting when the LUN capacity changes.

Device Mapper Operating Systems Support

Device mapper is supported for SLES 11 and RHEL 6.0 and later. All future updates of these OS versions are also supported. The following sections provide specific information on each of these operating systems.

Asymmetric Logical Unit Access (ALUA) with Linux Operating Systems

The ALUA feature is supported from Linux versions SLES11.1 and RHEL 6.1 onwards. You must download and install the Linux RPM packages to make use of this feature on SLES 11.1 and RHEL6.1. The rpm packages can be found on your storage vendor's website. Note that these packages are applicable only to SLES11.1 and RHEL 6.1. These packages are not required in SLES11.2, RHEL6.2 and subsequent releases for SLES and RHEL.

Understanding Device Handlers

DMMP uses different plug-ins called device handlers to manage failover and failback and to provide correct error handling. These device handlers are installed with the kernel during the installation of the operating system. The instructions for updating or configuring the device handlers are described in this guide.

- `scsi_dh_rdac`: Plug-in for DMMP that manages failover and failback through mode selects, manages error conditions, and allows the use of the ALUA feature, when enabled, on the storage array.
- `scsi_dh_alua`: Plug-in for DMMP for storage with Target Port Group Support (TPGS), which is a set of SCSI standards for managing multipath devices. This plugin manages failover and failback through the Set Target Port Group (STPG) command. This plug-in, however, is not supported in this release, and is not needed to run ALUA.

Installing the DMMP

All of the components required for DMMP are included on the installation media. By default, DMMP is disabled in SLES and RHEL. Complete the following steps to enable DMMP components on the host.

NOTE If you have not already installed the operating system, use the media supplied by your operating system vendor.

1. Use the procedures in the Setting Up the multipath.conf File section to update and configure the `/etc/multipath.conf` file.
2. On the command line, type **`chkconfig multipathd on`**.

   The multipathd daemon is enabled when the system starts again.
3. Enter the following commands to create the file `initramfs` and insert the command **`scsi_dh_rdac`** into the file `/etc/cmdline`. Note that this step is not required in RHEL7.

   ```
   #echo "rdloaddriver=scsi_dh_rdac" > /etc/cmdline

   #dracut --install '/etc/cmdline' -f /boot/initramfs-`uname -r`-rdac.img
   ```
4. Update the boot loader configuration file with the newly built `initrd` file.

   ATTENTION You must check the value for host type and, if necessary, change the setting of that value to ensure that ALUA can be enabled.
5. Do one of the following to verify and, if necessary, change the host type.
   - If you have hosts defined, go to step 6.
   - If you do not have hosts defined, right-click the default host group and then set the default host type to **Linux (DM-MP)**. Go to step 8.
6. In the SANtricity Storage Manager mappings view, right-click on the host and select **Change Host Operating System.**

7. Verify that the selected host type is **Linux (DMMP)**. If necessary, change the selected host type to **Linux (DMMP)**.
8. Reboot the host.

Migrating to the Linux DMMP Driver

This topic describes how to migrate to the Linux Device-Mapper Multipath (DMMP) driver from the RDAC driver.

Supported Operating Systems

Migrating from the RDAC failover driver to the Linux DMMP driver is supported on the following operating systems.

- RedHat Enterprise Linux 6.3, 6.4, 6.5
- SUSE Enterprise Linux Server 11.2, 11.3

Setting Up Persistent Block Device Naming

- You must use persistent device naming conventions before you can migrate to the Linux DMMP failover driver. Refer to the OS Administrator guide to setup persistent block device naming in the system.
- Device names will change based on the OS device discovery order. Using SCSI device names in mounting file systems or in boot loader configuration can lead to devices disappearing, unbootable system or kernel panic during reboot.
- You can view the file system labels and UUID's by running the following command.
  - #**lsblk -f** or #**lsblk -no UUID -f**

    ```
    91812c2b-7dfa-4d6b-854e-8c1ef2009f5d   +-sda1 swap [SWAP]

    f371aa77-07f1-4a28-91a9-2fb7a2912113   +-sda2 ext3 /
    ```
- After you’ve set up persistent device naming conventions, the file system table configuration (**/etc/fstab**) should reference all devices either by UUID or by Label as shown in the example below.

  ```
  #devices by UUID.

  UUID=88e584c0-04f4-43d2-ad33-ee9904a0ba32 /iomnt-sdb1 ext3 defaults 0 2

  UUID=2d8e23fb-a330-498a-bae9-5df72e822d38 /iomnt-sdc1 ext2 defaults 0 2

  UUID=43ac76fd-399d-4a40-bc06-9127523f5584 /iomnt-sdd1 xfs  defaults 0 2

  UUID=e30780e3-16ec-476a-92c2-c4324256af37 /iomnt-sde1 reiserfs defaults 0 2

  #devices by Label

  LABEL=db_vol /iomnt-vg1-lvol ext3 defaults 0 2

  LABEL=media_vol /iomnt-vg2-lvol xfs  defaults 0 2
  ```
- Make sure **symlinks** are created for all referenced devices under the **/dev/disk/by-uuid** directory or the /**dev/disk/by-label** directory.
- Verify that the root device in the boot loader configuration is referenced by either the UUID or Label as shown in the example below.

- **`linux /@/boot/vmlinuz-3.12.14-1-default root=UUID=e3ebb5b7-92e9-4928-aa33-55e2883b4c58`**or **`linux /@/boot/vmlinuz-3.12.14-1-default root=Label=root_vol`**

Migrating to the Linux DM-MP Driver

1. Un-install the RDAC driver.
   a. If RDAC is installed from the source, go to the RDAC source directory (typically the default location is under `/opt/StorageManager/`) and run the following command: **`#make uninstall`**
   b. Follow the suggestions as part of the uninstall script to remove the RDAC **`initrd`** image from the boot loader configuration (**`/boot/grub/menu.lst`**). The image is commonly named:**``mpp-`uname -r`.img``**
   c. If RDAC is installed from RPM, then run the following command to remove RDAC from the system: **`#rpm -e "RDAC RPM"`**
2. Install and configure the Linux DMMP multipath driver.

   Usually the DMMP packages are installed as part of base OS installation. If the DMMP packages are not installed, refer to the Installing the DMMP section for detailed instructions.
3. Change the host OS type in SANtricity Storage Manager.
   a. From the Array Management Window, select the storage array, and select **Host Mappings** > **Default Group** > **Change Default Host Operating System**.
   b. In the **New Host Type** list, select the host type **Linux (DMMP)**.
   c. Click **OK**.
4. Run the following command to implement the change to the host type.

   `#multipath -r`
5. Verify that no "ghost" paths are visible to the array on which the Linux (DMMP) host type is configured and verify that all path statuses are active and ready.

   `#multipath -ll`

```
mpatho (360080e50001b076d0000cd3251ef5eb0) dm-7 LSI     ,INF-01-00

size=5.0G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle'

hwhandler='1 rdac' wp=rw

|-+- policy='service-time 0' prio=14 status=active

| |- 5:0:1:15 sdag 66:0   active ready running

| - 6:0:1:15 sdbm 68:0   active ready running

`-+- policy='service-time 0' prio=9 status=enabled

  |- 5:0:0:15 sdq  65:0   active ready running

  - 6:0:0:15 sdaw 67:0   active ready running
```

6. Configure the HBA timeout values for the DMMP driver.
7. Verify that all the necessary **`SYMLINKS`** are created. (Prior to migration, the **`SYMLINKS`** pointed to RDAC virtual /**`dev/sdX devices`**.)

```
# ls -l /dev/disk/by-uuid

total 0
```

```
lrwxrwxrwx 1 root root 10 May 2 16:01 00011a47-8673-43d1-9810-b34ad3ee3af8 ->../../dm-6
lrwxrwxrwx 1 root root 11 May 2 16:01 2a12cf74-d494-4900-a98a-f1896f0766a2 ->../../dm-24
lrwxrwxrwx 1 root root 11 May 2 16:01 2d8e23fb-a330-498a-bae9-5df72e822d38 ->../../dm-16
lrwxrwxrwx 1 root root 11 May 2 16:01 43ac76fd-399d-4a40-bc06-9127523f5584 ->../../dm-20
lrwxrwxrwx 1 root root 10 May 2 16:01 5cc1188b-76bc-475c-b53f-400fe902ee6d ->../../dm-3
lrwxrwxrwx 1 root root 10 May 2 16:01 6dbbcbcf-2ece-4cea-a641-8fc1fe5f82db ->../../dm-8
lrwxrwxrwx 1 root root 11 May 2 16:01 88e584c0-04f4-43d2-ad33-ee9904a0ba32 -> ../../dm-17
lrwxrwxrwx 1 root root 10 May 2 16:01 91812c2b-7dfa-4d6b-854e-8c1ef2009f5d -> ../../sda1
lrwxrwxrwx 1 root root 11 May 2 16:01 d3faf0af-a547-403e-9506-323eea8ff1a4 ->../../dm-25
lrwxrwxrwx 1 root root 10 May 2 16:01 d5ca1d59-5486-4911-8d3c-c0a0629baa19 ->../../dm-9
lrwxrwxrwx 1 root root 11 May 2 16:01 e30780e3-16ec-476a-92c2-c4324256af37 ->../../dm-19
lrwxrwxrwx 1 root root 10 May 2 16:01 f371aa77-07f1-4a28-91a9-2fb7a2912113 ->../../sda2
lrwxrwxrwx 1 root root 10 May 2 16:01 f9d79e4b-ef7b-4c3c-a58b-c2064e488c13 ->../../dm-2
```

8. Verify the status of all the device paths.

   Using the appropriate commands from the **multipathd** and **multipath** utility, verify that all the device and path statuses are active and running as shown in the example below.

```
# multipathd show paths

hcil     dev  dev_t  pri dm_st  chk_st dev_st  next_check

5:0:0:0  sdb  8:16   14  active ready  running XXXXXXX... 14/20

5:0:0:1  sdc  8:32   9   active ready  running XXXXXXX... 14/20

5:0:0:10 sdl  8:176  9   active ready  running XXXXXXX... 14/20

5:0:0:11 sdm  8:192  14  active ready  running XXXXXXX... 14/20

#multipathd show maps

name    sysfs uuid

mpathaa dm-0  360080e50001b081000001b525362ff07

mpathj  dm-1  360080e50001b076d0000cd1a51ef5e6e

mpathn  dm-2  360080e50001b076d0000cd2c51ef5e9f

mpathu  dm-3  360080e50001b08100000044a51ef5e2b

#multipath -ll

mpatho (360080e50001b076d0000cd3251ef5eb0) dm-7 LSI     ,INF-01-00
```

```
size=5.0G features='4 queue_if_no_path pg_init_retries 50 retain_attached_hw_handle'

hwhandler='1 rdac' wp=rw

|-+- policy='service-time 0' prio=14 status=active

| |- 5:0:1:15 sdag 66:0   active ready running

| `- 6:0:1:15 sdbm 68:0   active ready running

`-+- policy='service-time 0' prio=9 status=enabled

  |- 5:0:0:15 sdq  65:0   active ready running

  `- 6:0:0:15 sdaw 67:0   active ready running
```

If any device path appears as failed, refer to the Troubleshooting the Device Mapper topic in this guide to troubleshoot the issue.

9. If present, verify the status of all the LVM devices.

   Run following commands to verify that all devices are referenced by "**mpath**" names rather than **sdX** device names.

```
#pvdisplay

  --- Physical volume ---

  PV Name               /dev/mapper/mpathx_part1

  VG Name               mpp_vg2

  PV Size               5.00 GiB / not usable 3.00 MiB

  Allocatable           yes

  PE Size               4.00 MiB

  Total PE              1279

  Free PE               1023

  Allocated PE          256

  PV UUID               v671wB-xgFG-CU0A-yjc8-snCc-d29R-ceR634

#vgdisplay

--- Volume group ---

  VG Name               mpp_vg2

  System ID

  Format                lvm2

  Metadata Areas        2
Metadata Sequence No  2
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               1
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               9.99 GiB
  PE Size               4.00 MiB
  Total PE              2558
  Alloc PE / Size       512 / 2.00 GiB
  Free  PE / Size       2046 / 7.99 GiB
  VG UUID               jk2xgS-9vS8-ZMmk-EQdT-TQRi-ZUNO-RDgPJz
#lvdisplay
  --- Logical volume ---
  LV Name                /dev/mpp_vg2/lvol0
  VG Name                mpp_vg2
  LV UUID                tFGMy9-eJhk-FGxT-XvbC-ItKp-BGnI-bzA9pR
  LV Write Access        read/write
  LV Creation host, time a7-boulevard, 2014-05-02 14:56:27 -0400
  LV Status              available
  # open                 1
  LV Size                2.00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
```

```
- currently set to     1024

Block device           253:24
```

10. If any issues are encountered, perform the appropriate file system checks on the devices.

Verifying that ALUA Support is Installed on the Linux OS

When you install or update host software and controller firmware to SANtricity version 10.83 and later and CFW version 7.83 and later and install DMMP on the Linux OS, support for ALUA is enabled. Installation must include the host software on the management station. Perform the following steps to verify that ALUA support is installed.

1. Perform one of the following actions to confirm that the host can see the LUNs that are mapped to it.
   - At the command prompt, type `SMdevices`. The appearance of active optimized or active non-optimized (rather than passive or unowned) in the output indicates that the host can see the LUNs that are mapped to it.
   - At the command prompt, type `multipath -ll`. If the host can see the LUNs that are mapped to it, both the path groups are displayed as active ready instead of active ghost.
2. Check the log file at `/var/log/messages` for entries similar to scsi 3:0:2:0: rdac: LUN 0 (IOSHIP).

   These entries indicate that the scsi_dh_rdac driver correctly recognizes ALUA mode. The keyword IOSHIP refers to ALUA mode. These messages are displayed when the devices are discovered in the system. These messages also might show in dmesg logs or boot logs.

Setting Up the multipath.conf File

The `multipath.conf` file is the configuration file for the multipath daemon, multipathd. The `multipath.conf` file overrides the built-in configuration table for multipathd. Any line in the file whose first non-white-space character is # is considered a comment line. Empty lines are ignored.

Example `multipath.conf` are available in the following locations:

- For SLES, `/usr/share/doc/packages/multipath-tools/multipath.conf.synthetic`
- For RHEL, `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf`

All the lines in the sample `multipath.conf` file are commented out. The file is divided into five sections:

- **defaults** – Specifies all default values.
- **blacklist** – All devices are blacklisted for new installations. The default blacklist is listed in the commented-out section of the `/etc/multipath.conf` file. Blacklist the device mapper multipath by WWID if you do not want to use this functionality.
- **blacklist_exceptions** – Specifies any exceptions to the items specified in the section blacklist.
- **devices** – Lists all multipath devices with their matching vendor and product values.
- **multipaths** – Lists the multipath device with their matching WWID values.

In the following tasks, you modify the default, blacklist and devices sections of the `multipath.conf` file. Remove the initial # character from the start of each line you modify.

Updating the Blacklist Section

With the default settings, UTM LUNs might be presented to the host. I/Os operations, however, are not supported on UTM LUNs. To prevent I/O operations on the UTM LUNs, add the vendor and product information for each UTM LUN to the blacklist section of the `/etc/multipath.conf` file. The entries should follow the pattern of the following example.

```
blacklist {

       device {

              vendor "*"

              product "Universal Xport"

       }

}
```

Setting Up Multipath.conf to Blacklist QS/QD-Series Devices

If you need to run the RDAC failover solution that comes with the QS/QD-Series storage management software, and the Device-Mapper Multipath for storage from another vendor, you must update the **`multipath.conf`** file (`/etc/multipath.conf`) to blacklist all of the QS/QD-Series storage array volumes.

By vendor and product id:

```
blacklist {

device  {

       vendor "NETAPP"

       product "INF-01-00"

        }

}
```

You can also blacklist each volume by worldwide id, for individual volumes:

```
blacklist {

          device {

            wwid"360080e50001be4880000217e51e69e4f"

          }

}
```

Note that there should only be one blacklist block in **`multipath.conf`**, but it can contain multiple device entries.

Restart the multipathd service (**`service multipathd restart`**) for the changes to take effect.

Updating the Devices Section of the multipath.conf File

If your host is running RHEL 6.5 or SLES 11.3 or any prior release to RHEL 6.5 or SLES 11.3, update the /etc/multipath.conf file as described below. If you are using a later release, simply create an empty /etc/multipath.conf file. When you create an empty **multipath.conf** file, the system automatically applies all the default configurations, which includes supported values for QS/QD-Series devices.

Devices section

The following example shows part of the devices section in the /etc/multipath.conf file. The the example shows the vendor ID as NETAPP or LSI and the product ID as INF-01-00. Modify the devices section with product and vendor information to match the configuration of your storage array. If your storage array contains devices from more than one vendor, add additional device blocks with the appropriate attributes and values under the devices section.

NOTE Update the devices section of the multipath.conf file only if your host is running RHEL 6.5 or SLES 11.3 or any prior release to RHEL 6.5 or SLES 11.3.

```
devices {

  device {

    vendor           "(LSI|NETAPP)"

    product          "INF-01-00"

    path_grouping_policy  group_by_prio

    prio             rdac

    path_checker     rdac

    hardware_handler "1 rdac"

    failback         immediate

    features         "2 pg_init_retries 50"

    no_path_retry    30

         }

}
```

Table 27. Attributes and Values in the multipath.conf File

Attribute	Parameter Value	Description
path_grouping_policy	**group_by_prio**	The path grouping policy to be applied to this specific vendor and product storage.

Attribute	Parameter Value	Description
prio	**rdac**	The program and arguments to determine the path priority routine. The specified routine should return a numeric value specifying the relative priority of this path. Higher numbers have a higher priority.
path_checker	**rdac**	The method used to determine the state of the path.
hardware_handler	**"1 rdac"**	The hardware handler to use for handling device-specific knowledge.
failback	**immediate**	A parameter to tell the daemon how to manage path group failback. In this example, the parameter is set to 10 seconds, so failback occurs 10 seconds after a device comes online. To disable the failback, set this parameter to **manual**. Set it to **immediate** to force failback to occur immediately. When clustering or shared LUN environments are used, set this parameter to **manual**.
features	**"2 pg_init_retries 50"**	Features to be enabled. This parameter sets the kernel parameter pg_init_retries to **50**. The pg_init_retries parameter is used to retry the mode select commands.
no_path_retry	**30**	Specify the number of retries before queuing is disabled. Set this parameter to **fail** for immediate failure (no queuing). When this parameter is set to **queue**, queuing continues indefinitely. The amount of time is equal to the parameter value multiplied by the **polling_interval** (usually 5), for example,150 seconds for a **no_path_retry** value of 30.

Setting Up DM-MP for Large I/O Blocks

When a single I/O operation request a block larger than 512 KB, this is considered to be a large block. You must tune certain parameters for a device that uses Device Mapper Multipath (DMMP) in order for the device to perform correctly with large I/O blocks. Parameters are usually defined in terms of blocks in the kernel, and are shown in terms of kilobytes to the user. For a normal block size of 512 bytes, simply divide the number of blocks by 2 to get the value in kilobytes. The following parameters affect performance with large I/O blocks:

- max_hw_sectors_kb (RO) - This parameter sets the maximum number of kilobytes that the hardware allows for request.

- `max_sectors_kb (RW)` - This parameter sets the maximum number of kilobytes that the block layer allows for a file system request. The value of this parameter must be less than or equal to the maximum size allowed by the hardware. The kernel also places an upper bound on this value with the BLK_DEF_MAX_SECTORS macro. This value varies from distribution to distribution, for example, it is 1024 on RHEL6.3, 2048 on SLES11 SP2.
- `max_segments (RO)` - This parameter enables low level driver to set an upper limit on the number of hardware data segments in a request. In the HBA drivers, this is also known as `sg_tablesize`.
- `max_segment_size (RO)` - This parameter enables low level driver to set an upper limit on the size of each data segment in an I/O request in bytes. If clustering is enabled on the low level driver it is set to 65536 or it is set to system `PAGE_SIZE` by default, which is typically 4K. The maximum I/O size is determined by the following:

  ```
  MAX_IO_SIZE_KB = MIN(max_sectors_kb, (max_segment_size * max_segments)/1024)
  ```

  where `PAGE_SIZE` is architecture independent. It is 4096 for x86_64.

1. Set the value of the `max_segments` parameter for the respective HBA driver as load a time module parameter.

   The following table lists HBA drivers which provide module parameters to set the value for `max_segments`.

HBA	Module Parameter
LSI SAS (mpt2sas)	`max_sgl_entries`
Emulex (lpfc)	`lpfc_sg_seg_cnt`
Infiniband (ib_srp)	`cmd_sg_entries`
Brocade (bfa)	`bfa_io_max_sge`

2. If supported by the HBA, set the value of `max_hw_sectors_kb` for the respective HBA driver as a load time module parameter. This parameter is in sectors and is converted to kilobytes.

HBA	Parameter	How to Set
LSI SAS (mpt2sas)	`max_sectors`	Module parameter
Infiniband (ib_srp)	`max_sect`	Open **`/etc/srp_daemon.conf`** and add **`"a max_sect=<value>"`**
Brocade (bfa)	`max_xfer_size`	Module parameter

3. On the command line, enter the command `echo` *N* `>/sys/block/`*sd device name*`/queue/max_sectors_kb` to set the value for the `max_sectors_kb` parameter for all physical paths for dm device in sysfs. In the command, *N* is an unsigned number less than the `max_hw_sectors_kb` value for the device; *sd device name* is the name of the sd device.
4. On the command line, enter the command `echo` *N* `>/sys/block/`*dm device name*`/queue/max_sectors_kb` to set the value for the `max_sectors_kb` parameter for all dm device in sysfs. In the command, *N* is an unsigned number less than the `max_hw_sectors_kb` value for the device; *dm device name* is the name of the dm device represented by dm-X.

Using the Device Mapper Devices

Multipath devices are created under `/dev/` directory with the prefix `dm-`. These devices are the same as any other block devices on the host. To list all of the multipath devices, run the **`multipath –ll`** command.

The following example shows system output from the **`multipath –ll`** command for one of the multipath devices.

```
mpathg (360080e50001be48800001c9a51c1819f) dm-8 NETAPP,INF-01-00

size=30G features='3 queue_if_no_path pg_init_retries 50' hwhandler='1 rdac' wp=rw
```

```
|-+- policy='round-robin 0' prio=14 status=active

| |- 16:0:0:4  sdau 66:224 active ready  running

| `- 15:0:0:4  sdbc 67:96  active ready  running

`-+- policy='round-robin 0' prio=9 status=enabled

|- 13:0:0:4  sdat 66:208 active ready  running

`- 14:0:0:4  sdbb 67:80  active ready  running
```

In this example, the multipath device nodes for this device are `/dev/mapper/mpathp` and `/dev/dm-0`. This example shows how the output should appear during normal operation. The lines beginning with "`policy=`" are the path groups. There should be one path group for each controller. The path group currently being used for I/O access will have a status of `active`. To verify that ALUA is enabled, all `prio` values should be greater than 8, and all paths should show `active ready` as their status.

The following table lists some basic options and parameters for the `multipath` command.

Table 28. Options and Parameters for the multipath Command

Command	Description
`multipath -h`	Prints usage information
`multipath`	With no arguments, attempts to create multipath devices from disks not currently assigned to multipath devices
`multipath -ll`	Shows the current multipath topology from all available information, such as the sysfs, the device mapper, and path checkers
`multipath -11 map`	Shows the current multipath topology from all available information, such asthe sysfs, the device mapper, and path checkers
`multipath -f map`	Flushes the multipath device map specified by the map option, if the map is unused
`multipath -F`	Flushes all unused multipath device maps

How to Use Partitions on DM Devices

Multipath devices can be partitioned like any other block device. When you create a partition on a multipath device, device nodes are created for each partition. The partitions for each multipath device have a different dm- number than the raw device.

For example, if you have a multipath device with the `WWID 3600a0b80005ab177000017544a8d6b9c` and the user friendly name `mpathb`, you can reference the entire disk through the following path:

```
/dev/mapper/mpathb
```

If you create two partitions on the disk, they will be accessible through the following path:

```
/dev/mapper/mpathbp1

/dev/mapper/mpathbp2.
```

If you do not have user friendly names enabled, the entire disk will be accessible through the following path:

```
/dev/mapper/3600a0b80005ab177000017544a8d6b9c
```

And the two partitions are accessible through the following path:

```
/dev/mapper/3600a0b80005ab177000017544a8d6b9cp1
```

```
/dev/mapper/3600a0b80005ab177000017544a8d6b9cp2
```

Troubleshooting the Device Mapper

Table 29. Troubleshooting the Device Mapper

Situation	Resolution
Is the multipath daemon, multipathd, running?	At the command prompt, enter the command: **`#service multipathd status`**.
Why are no devices listed when you run the **`multipath -ll`** command?	At the command prompt, enter the command: `#cat /proc/scsi/scsi`. The system output displays all of the devices that are already discovered. Verify that the `multipath.conf` file has been updated with proper settings. You can check the running configuration with the **`multipathd show config`** command.

Failover Drivers for the AIX/PowerVM Operating System

Multipath I/O (MPIO) is the supported failover driver for the AIX/ PowerVM operating system on the QS/QD-Series systems.The MPIO driver has basic failover features such as fault-tolerance and performance monitoring.

The AIX / PowerVM operating system has three types of failover drivers, which include:

- MPIO
- SDDPCM
- RDAC

Only the MPIO driver is supported with the QS/QD-Series systems.

About the AIX/PowerVM Failover Driver

The primary function of the MPIO driver is to appropriately choose the physical paths on which to route I/O. In the event of a path loss, the MPIO driver will re-route I/O to other available paths (failover) with minimal interruption and no user interaction.

The MPIO driver allows a device to be detected through one or more physical connections or path. The MPIO capable device driver can control more than one type of target device. The interaction of different components such as the Device Driver capability, PCM, and Object Data Management (ODM) make up the MPIO solution.

Before an QS/QD-Series device can take advantage of the MPIO driver, the predefined attributes in the ODM must be modified to support detection, configuration, and management of the QS/QD-Series.

Listing the Device Driver Version (MPIO)

NOTE Where you enter the following commands depends on whether you are using the NPIV configuration or the vSCSI PowerVM configuration.

To list the MPIO device driver version, run the command below:

```
# lslpp -l devices.common.IBM.mpio.rte
```

To list the MPIO device according to its respective storage on the QS/QD-Series device, run the command below:

```
# mpio_get_config -l hdiskxx (Where "xx" represent the hdisk number E.g : "hdisk5")
```

To list all path information, run the command below:

```
# lspath
```

IMPORTANT The **`mpio_get_config -Av`** command is not supported on QS/QD-Series devices with the AIX/ PowerVM operating system.

Validating Object Data Management (ODM)

ODM is an integral part of device configuration on AIX/PowerVM. ODM contains the default values for the MPIO driver that must be modified so the MPIO driver can take advantage of your QS/QD-Series devices.

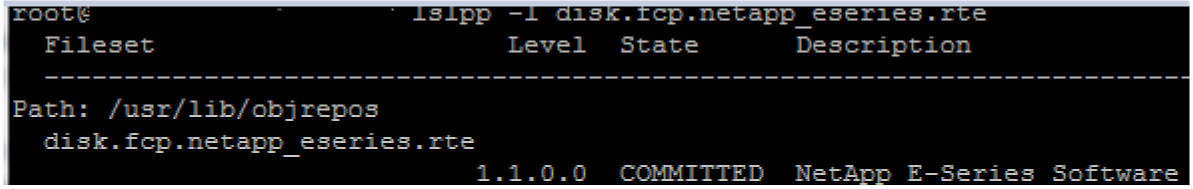
A good understanding of ODM is critical for solving AIX device issues such as boot up, I/O transfer error, and device management. To make sure that the modifications are automatically made to ODM, install the SANtricity Storage Manager for AIX.

NOTE Refer to the *SANtricity® Storage Manager 11.20 Software Installation Reference Guide* and the *SANtricity® Storage Manager 11.20 System Upgrade Guide* for more information about the ODM entry installation.

To validate the ODM, run the following command:

```
# lslpp -l disk.fcp.netapp_eseries.rte
```

The expected result is shown in the following example:



```
root@                        lslpp -l disk.fcp.netapp_eseries.rte
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  disk.fcp.netapp_eseries.rte
                             1.1.0.0  COMMITTED  NetApp E-Series Software
```

82013-01

NOTE Where this command is performed depends on whether you are using the NPIV configuration or the vSCSI PowerVM configuration.

Understanding the Recommended AIX Settings and HBA Settings

Please check your AIX servers for the recommended default settings and the HBA settings.

NOTE Where these commands are run depend on whether you are using the NPIV configuration or the vSCSI PowerVM configuration.

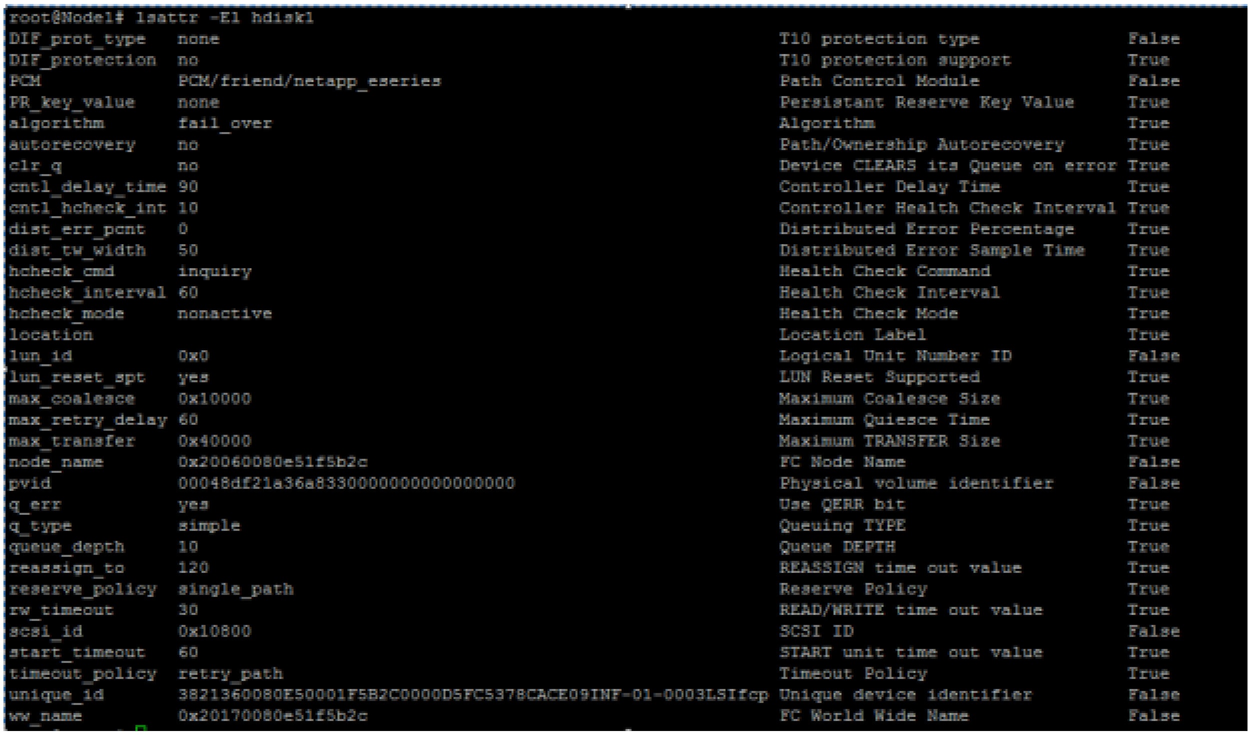
Checking the AIX default settings

Run the following command to check the default settings for AIX.

```
# lsattr -El hdiskxx
```

where **xx** is the hdisk number.

The expected output is similar to that shown in the following example.



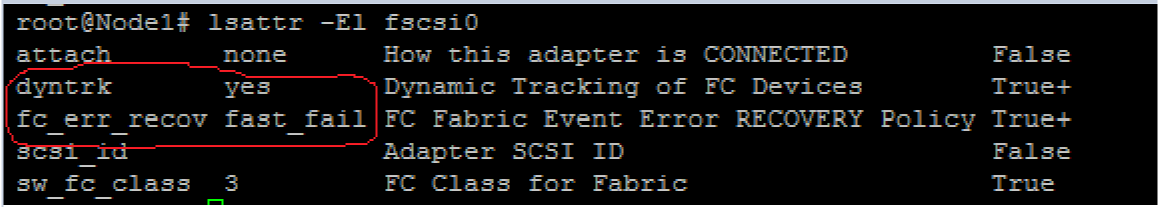
```
root@Node1# lsattr -El hdisk1
DIF_prot_type   none                                                     T10 protection type              False
DIF_protection  no                                                       T10 protection support           True
PCM             PCM/friend/netapp_eseries                                Path Control Module              False
PR_key_value    none                                                     Persistant Reserve Key Value     True
algorithm       fail_over                                                Algorithm                        True
autorecovery    no                                                       Path/Ownership Autorecovery      True
clr_q           no                                                       Device CLEARS its Queue on error True
cntl_delay_time 90                                                       Controller Delay Time            True
cntl_hcheck_int 10                                                       Controller Health Check Interval True
dist_err_pcnt   0                                                        Distributed Error Percentage     True
dist_tw_width   50                                                       Distributed Error Sample Time    True
hcheck_cmd      inquiry                                                  Health Check Command             True
hcheck_interval 60                                                       Health Check Interval            True
hcheck_mode     nonactive                                                Health Check Mode                True
location                                                                 Location Label                   True
lun_id          0x0                                                      Logical Unit Number ID           False
lun_reset_spt   yes                                                      LUN Reset Supported              True
max_coalesce    0x10000                                                  Maximum Coalesce Size            True
max_retry_delay 60                                                       Maximum Quiesce Time             True
max_transfer    0x40000                                                  Maximum TRANSFER Size            True
node_name       0x20060080e51f5b2c                                       FC Node Name                     False
pvid            00048df21a36a8330000000000000000                         Physical volume identifier       False
q_err           yes                                                      Use QERR bit                     True
q_type          simple                                                   Queuing TYPE                     True
queue_depth     10                                                       Queue DEPTH                      True
reassign_to     120                                                      REASSIGN time out value          True
reserve_policy  single_path                                              Reserve Policy                   True
rw_timeout      30                                                       READ/WRITE time out value        True
scsi_id         0x10800                                                  SCSI ID                          False
start_timeout   60                                                       START unit time out value        True
timeout_policy  retry_path                                               Timeout Policy                   True
unique_id       3821360080E50001F5B2C0000D5FC5378CACE09INF-01-0003LSIfcp Unique device identifier         False
ww_name         0x20170080e51f5b2c                                       FC World Wide Name               False
```

82013-00

Checking the HBA Settings

Most of the default settings are set by the ODM except for the following two HBA settings:

```
dyntrk=yes

fc_err_recov=fast_fail
```



```
root@Node1# lsattr -El fscsi0
attach       none      How this adapter is CONNECTED         False
dyntrk       yes       Dynamic Tracking of FC Devices        True+
fc_err_recov fast_fail FC Fabric Event Error RECOVERY Policy True+
scsi_id                Adapter SCSI ID                       False
sw_fc_class  3         FC Class for Fabric                   True
```

82013-02

Run the following command to check the default settings for HBA.

```
# lsattr -El fscsixx
```

where **xx** is the fscsi number.

You must manually set the **dyntrk** and the **fc_err_recov** HBA settings. Run the following command to change these settings:

```
# chdev -l fscsix -a dyntrk=yes -a fc_err_recov=fast_fail -P
```

where **xx** is the fscsi number.

You can also run the following script to change the HBA settings:

```
#!/usr/bin/ksh

# This script changes fscsi device attributs from delayed_fail to fast_fail (fast_fail ON)

and dyntrk from no to yes

lscfg | grep fscsi | cut -d' ' -f2 | while read line

do

chdev -l $line -a fc_err_recov=fast_fail

lsattr -El $line | grep fc_err_recov

chdev -l $line -a dyntrk=yes

lsattr -El $line | grep dyntrk

done

echo "YOU MUST RESCAN THE SYSTEM NOW FOR THE CHANGES TO TAKE EFFECT"
```

Enabling the Round-Robin Algorithm

The ODM I/O algorithm is set to failover by default. Quantum recommends using round-robin to achieve optimal performance. In addition, set the **`Reserve_policy`** parameter to "**`no_reserve`**". This change will allow I/O to be distributed across all enabled adapters to the owning controller ports.

Checking the Algorithm Default Settings

NOTE Where these commands are run depend on whether you are using the NPIV configuration or the SCSI PowerVM configuration.

Run the following command to check the default settings for the ODM I/O algorithm.

```
# lsattr -El hdiskxx
```

where **xx** is the hdisk number.

Changing the Round-Robin Algorithm

NOTE You have to manually set up the algorithm and the **`reserve_policy`** to "**`round_robin`**" and "**`no_reserve`**" on QS/QD-Series devices.

For each QS/QD-Series hdisk in your configuration, run the following **chdev** command to change the algorithm. #

```
chdev -l hdiskxx -a 'algorithm=round_robin reserve_policy=no_reserve'
```

where **xx** is your hdisk number.

If the algorithm setting was successfully changed, you will see a message string similar to the following example.

```
# chdev -l hdisk1 -a 'algorithm=round_robin reserve_policy=no_reserve
```

If you have file systems or volume groups on the hdisk, the **chdev** command will fail as show in the following example.

```
E.g (algorithm setting failed) :

# chdev -l hdisk5 -a 'algorithm=round_robin reserve_policy=no_reserve'

Error (The device has a Filesystem or is part of a volume group):
```

Troubleshooting the MPIO Device Driver

Problem	Recommended Action

Why are no paths listed when I run **`lspath`**?	Make sure the ODM is installed. At the command prompt, enter the following command:
	`# lslpp -l disk.fcp.netapp_eseries.rte`
	Check the HBA settings and the failover settings.
	To rescan, enter the following command:
	`# cfgmgr`

Why are no devices listed when I run the**`mpio_get_config -Av`** command?	This command will not work on AIX/PowerVM with QS/QD-Series. Instead, run the following command: `# mpio_get_config -l hdiskxx` where **`hdiskxx`** represents the MPIO device on the QS/QD-Series storage system.

Failover Drivers for the Solaris Operating System

MPxIO is the supported failover driver for the Solaris operating system.

Solaris OS Restrictions

SANtricity Storage Manager no longer supports or includes RDAC for the following Solaris operating systems:

- Solaris 10
- Solaris 11

MPxIO Load Balancing Policy

The load-balancing policy that you can choose for the Solaris MPxIO multi-path driver is the Round Robin with subset policy.

The round robin with subset I/O load-balancing policy routes I/O requests, in rotation, to each available data path to the controller that owns the volumes. This policy treats all paths to the controller that owns the volume equally for I/O activity. Paths to the secondary controller are ignored until ownership changes. The basic assumption for the round robin with subset I/O policy is that the data paths are equal. With mixed host support, the data paths might have different bandwidths or different data transfer speeds.

Enabling MPxIO on the Solaris 10 OS

MPxIO is included in the Solaris 10 OS. Therefore, MPxIO does not need to be installed. It only needs to be enabled.

NOTE MPxIO for iSCSI is enabled by default.

1. To enable MPxIO for a specific protocol, run one of the following commands:
   - To enable FC drives, run the **`stmsboot -D fp -e`** command.
   - To enable 3-GB SAS drives, run the **`stmsboot -D mpt -e`** command.
   - To enable 6-GB SAS drives, run the **`stmsboot -D mpt_sas -e`** command.

   To find the correct parent and port numbers, look at the device entry for the internal drives, found in the `/dev/dsk/path`.
2. Reboot the system.
3. To enable or disable MPxIO on specific drives port, add a line similar to the following to the **`/kernel/drv/fp.conf`** Fibre Channel port driver configuration file:

   Enable

   ```
   name="fp" parent="/pci@8,600000/SUNW,qlc@2" port=0 mpxio-disable="no";
   ```

   Disable

```
name="fp" parent="/pci@8,600000/SUNW,qlc@2" port=0 mpxio-disable="yes";
```

To find the correct parent and port numbers, look at the device entry for the internal drives, found in the `/dev/dsk/path`.

4. To globally enable or disable MPxIO, run one of the following commands:

   Enable

   ```
   # stmsboot -e
   ```

   Disable

   ```
   # stmsboot -d
   ```

Enabling MPxIO on the Solaris 11 OS

MPxIO is included in the Solaris 11 OS. Therefore, MPxIO does not need to be installed. It only needs to be enabled.

NOTE MPxIO for the x86 architecture is, by default, enabled for the Fibre Channel (FC) protocol.

1. To enable MPxIO for FC drives, run the following command: **`stmsboot -D fp -e`**
2. Reboot the system.

Configuring Failover Drivers for the Solaris OS

Use the default settings for all Solaris OS configurations.

Frequently Asked Questions about Solaris Failover Drivers

Table 30. Frequently Asked Questions about Solaris Failover Drivers

Question	Answer

Where can I find MPxIO-related files?	You can find MPxIO-related files in these directories:
	`/etc/`
	`/kernel/drv`

Where can I find data files?	You can find data files in these directories: `/var/opt/SM`
Where can I find the command line interface (CLI) files?	You can file CLI files in this directory: `/usr/sbin`
Where can I find the bin files?	You can find the bin files in the `/usr/sbin` directory.

Where can I find device files?	You can find device files in these directories:
	`/dev/rdsk`
	`/dev/dsk`

Question	Answer

Where can I find the SANtricity Storage Manager files?	You can find the SANtricity Storage Manager files in these directories:
	`/opt/SMgr`
	`/opt/StorageManager`

Where can I get a list of storage arrays, their volumes, LUNs, WWPNs, preferred paths, and owning controller?	Use the SMdevices utility, which is located in the `/usr/bin` directory. You can run the `SMdevices` utility from any command prompt.
How can I see whether volumes have been added?	Use the **`devfsadm`** utility to scan the system. Then run the `mpathadm list lu` command to list all volumes and their paths. If you still cannot see any new volumes, either reboot the host and run the `mpathadm list lu` command again, or use the **`SMdevices`** utility. The `mpathadm list lu` command works only if MPxIO is enabled. As an alternative, list this information by entering the **`luxadm probe`** command.
How do I find which failover module manages a volume in Solaris 11?	Check the host log messages for the volume. Storage arrays with Asymmetric Logical Unit Access (ALUA) are managed by the `f_tpgs` module. Storage arrays with earlier version of firmware are managed by the `f_asym_lsi` module. As an alternative, list this information by selecting one of the devices/LUN you would like to check and then enter the following command: `#` **`mpathadm list lu`**. For example: `#` **`mpathadm show lu /dev/rdsk/c0t60080E5000290B1C0000091B536FEA47d0s2`**
How can I determine the failover support for my device?	Use the following command to list the vendors VID. `#` **`mpathadm show mpath-support libmpscsi_vhci.so`**. If the VID is not displayed with the command shown above, then **`f_tpgs`** will be used (if the target supports TPGS).
Where can I find the backup of the .conf files?	All files are saved in `/etc/mpxio/` in a file name formed by concatenating the original file name, the timestamp and an indication of whether the file was enabled or disabled as shown in the example below: **`fp.conf.enable.20140509_1328`**

Installing ALUA Support for VMware Versions ESX4.1U3, ESXi5.0U1, and Subsequent Versions

Starting with ESXi5.0 U1 and ESX4.1U3, VMware will automatically have the claim rules to select the VMW_SATP_ALUA plug-in to manage storage arrays that have the target port group support (TPGS) bit enabled. All arrays with TPGS bit disabled are still managed by the VMW_SATP_LSI plug-in.

1. Make sure that the host software on the management station is upgraded to version 10.86.
2. Upgrade the controllers in the storage array to controller firmware version 7.86 and the corresponding NVSRAM version.
3. From host management client, verify that the host OS type is set to *`VMWARE`*. Starting with storage management software version 10.84, the *`VMWARE`* host type will have the ALUA and TPGS bits enabled by default.
4. Use one of the following command sequences to verify that the TPGS/ALUA enabled devices are claimed by the VMW_SATP_ALUA plug-in.
   - For ESX4.1, enter the command `#esxcli nmp device list` on the command line of the host. Check that the output shows `VMW_SATP_ALUA` as the value of `Storage Array Type` for every storage array whose host software level is 10.83 or higher. Storage arrays with lower level host software show `VMW_SATP_LSI` as the value of `Storage Array Type`.
   - For ESXi5.0, enter the command `#esxcli storage nmp device list` on the command line of the host. Check that the output shows `VMW_SATP_ALUA` as the value of `Storage Array Type` for every storage array whose host software level is 10.83 or higher. Storage arrays with lower level host software show `VMW_SATP_LSI` as the value of `Storage Array Type`.