SANtricity® Storage Manager 11.25

# Multipath Drivers Guide

April 2017 | 215-09822_B0
doccomments@netapp.com

NetApp®

# Contents

# Deciding whether to use this guide

The guide describes how to install and configure the supported multipath 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 multipath driver on a host system installed with Windows, Linux, AIX, Solaris, or VMware operating systems.
- Configure multiple physical paths (multipaths) to storage in order to implement a standard installation and configuration using best practices.

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.

**Note:** This guide does not provide information about device-specific information, all the available configuration options, or a lot of conceptual background for the tasks.

### 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 *NetApp E-Series and EF-Series Systems Documentation Center*.

# Summary of changes for Multipath Drivers Guide

You can find information about changes to the multipath drivers related to the latest release of the SANtricity software.

The following describes new and updated topics in this document.

In **Overview of multipath drivers**:

In *Supported multipath drivers* on page 9:

- Added the ATTO Cluster/All OS host type.
- The MPP/RDAC multipath driver is no longer supported for Linux operating systems. Users should ensure they are using the DM-MP multipath driver.
- The MPxIO (non-TPGS) multipath driver is no longer supported for Solaris operating systems. Users should ensure they are using the MPxIO (TPGS/ALUA) multipath driver
- The RDAC multipath driver is no longer supported for ONTAP. Users should ensure they are using the native ALUA multipath driver.

*Single-Host configuration* on page 10 has been updated to provide an updated configuration example.

*Direct connect and fabric connect configurations* on page 11 has been updated to provide an updated configuration example.

In **Multipath drivers for the Linux operating system**:

*Multipath drivers for the Linux operating system* on page 43 cites that the MPP/RDAC multipath driver is no longer supported.

*Known limitations and issues of the device mapper multipath* on page 44 has been updated to provide more information.

*Setting up the multipath.conf file* on page 51 now cites the order of operations for parameters.

*Updating the devices section of the multipath.conf file* on page 52 has added new parameters.

*Rescanning devices with the DM-MP multipath driver* on page 56 has been added to help you rescan SCSI devices to work with the new DM-MP multipath driver.

In **Multipath drivers for the AIX/PowerVM operating system**:

*Understanding the recommended AIX settings and HBA settings* on page 59 has been updated to show that the `algorithm` default has been changed from "fail_over" to "round_robin" and the `reserve_policy` default has been changed form "single_path" to "no_reserve."

*Enabling the failover algorithm* on page 61 has been updated to show how to manually set the algorithm to "fail_over," because it now defaults to "round_robin."

In **Multipath drivers for the Solaris operating system**:

*Enabling MPxIO on the Solaris 10 and 11 OS* on page 63 describes how to enable the MPxIO multipath driver on both Solaris 10 and Solaris 11.

*Editing the sd.conf file and the ssd.conf file for TPGS support in Solaris 10* on page 64 has been added to provide updates to the Solaris 10 systems so failover with TPGS works correctly.

In **Multipath drivers for the VMware operating system and upgrade instructions**:

*Multipath drivers for the VMware operating system and upgrade instruction* on page 67 has been updated to describe how multipath software for the VMware vSphere ESXi is handled natively by the operating system through the Native Multipathing Plugin (NMP), while providing required updates for ESXi 5.5 hosts prior to U2 and all ESXi 5.1 versions.

# Overview of multipath drivers

Multipath 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 multipath driver automatically reroutes input/output (I/O) to the good path. Multipath drivers help the hosts to continue to operate without interruption when the path fails.

Multipath 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 (failover).
- 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 asymmetric logical unit access (ALUA) mode of operation has changed.
- They provide load balancing between available paths.

## Multipath 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 multipath driver software, which controls data flow to the controller pairs. This software tracks the current status of the connections and can perform the failover.

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 multipath 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

To ensure reliable operation, you must verify that your configuration is supported.

The information in this table is intended to provide general guidelines. Refer to the *NetApp Interoperability Matrix Tool* for compatibility information for specific HBA, multipath driver, OS level, and controller-drive tray support. The drivers are listed by operating system, starting with those described in this document, followed by all other options in alphabetical order.

| Operating System | Multipath Driver | Recommended Host Type |
|---|---|---|
| Windows Server | MPIO with NetApp E-Series DSM (with ALUA support) | Windows or Windows Clustered |
| Windows | ATTO Multipath Director | Windows ATTO<br>**Note:** You must use ATTO FC HBAs. |
| Windows | ATTO Multipath Director and clustered/parallel file system | ATTO Cluster/All OS<br>**Note:** You must use ATTO FC HBAs and clustered/parallel file systems. |
| Linux | DM-MP with RDAC handler (with ALUA support) | Linux (DM-MP) |
| Linux | ATTO Multipath Director | Linux (ATTO)<br>**Note:** You must use ATTO FC HBAs. |
| Linux | ATTO Multipath Director and clustered/parallel file system | ATTO Cluster/All OS<br>**Note:** You must use ATTO FC HBAs and clustered/parallel file systems. |
| Linux | Symantec/Veritas Storage Foundation Multipath Driver | Linux (Symantec Storage Foundations) |
| Linux | VxDMP | Linux (Symantec Storage Foundation) |
| AIX<br>VIOS | Native MPIO | AIX MPIO |
| Solaris | MPxIO (non-TPGS) | Solaris Version 10 or earlier |
| Solaris | MPxIO (TPGS/ALUA) | Solaris Version 11 or later |
| VMware | Native Multipathing Plugin (NMP) with VMW_SATP_ALUA Storage Array Type Plugin (SATP) | VMware |
| HP-UX | Native TPGS/ALUA | HP-UX |
| Mac | ATTO Multipath Director | Mac OS |

| Operating System | Multipath Driver | Recommended Host Type |
|---|---|---|
| Mac | ATTO Multipath Director and clustered/parallel file system | ATTO Cluster/All OS<br>**Note:** You must use ATTO FC HBAs, and clustered/parallel file systems. |
| ONTAP | Native ALUA | Data ONTAP (ALUA) |
| n/a | n/a | Factory Default<br>**Note:** Default host type that should only be used during initial configuration. The host type should be set to another selection that is configured for the host OS and multipath driver being used. |

When you select either the **Typical (Full Installation)** option or the **Custom** installation option through the SMagent package, the host context agent 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 automatically define the host and the associated host ports, and set the host type. The host context agent sends the host topology to the storage controllers only once, and any subsequent changes made in SANtricity Storage Manager is 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, select the host, and 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**.

# Multipath configuration diagrams

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

- Single-host configuration
- Direct connect and fabric connect configurations

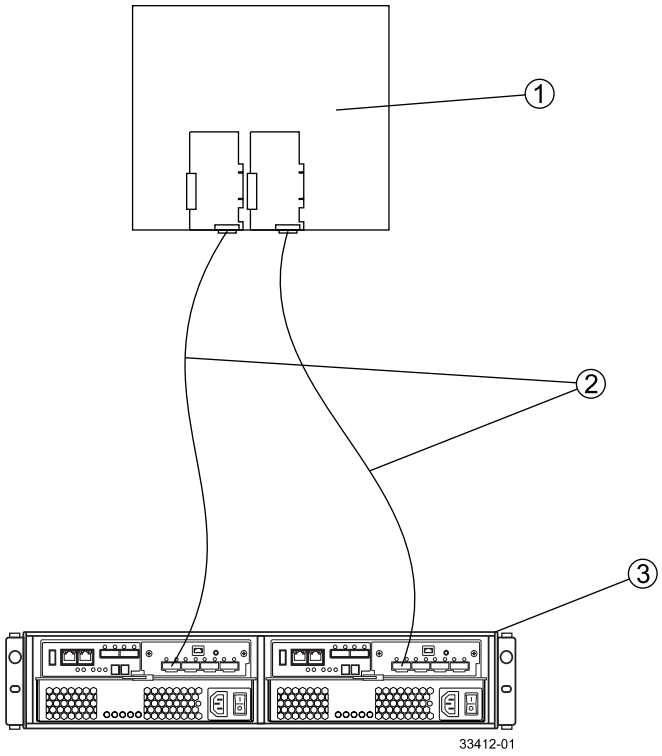
This section also describes how the storage management software supports redundant controllers.

## Single-Host configuration

In a single-host configuration, the host system contains two host bus adapters (HBAs), with a port on each HBA connected to different 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.




1. Host System with Two SAS, Fibre Channel, iSCSI, or InfiniBand Host Bus Adapters
2. SAS, Fibre Channel, iSCSI, or InfiniBand Connection – The Network Protocol Connection Might Contain One or More Switches
3. Storage Array with Two Controllers

## Direct connect and fabric connect configurations

In a direct connect or fabric connect configuration, two host systems are each connected by two connections to both of the controllers in a storage array. SANtricity Storage Manager, including multipath 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.

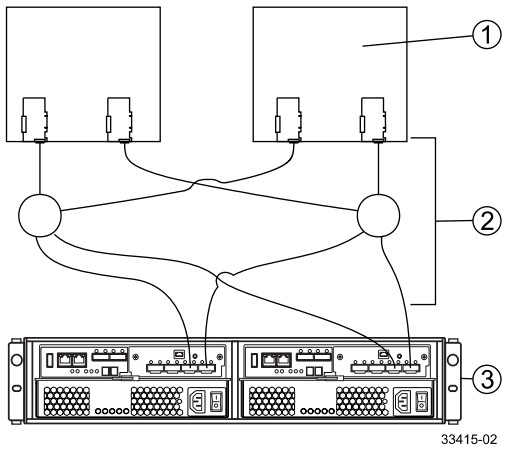
In either a direct connect or fabric connect configuration, each host has visibility to both controllers, all data connections, and all configured volumes in a storage array.

The following conditions apply to these both direct connect and fabric connect configurations:

- Both hosts must have the same operating system version installed.

  **Note:** For VMware this condition does not apply because you can cluster ESXi 5.0, 5.1, 5.5, and 6.0 releases together.

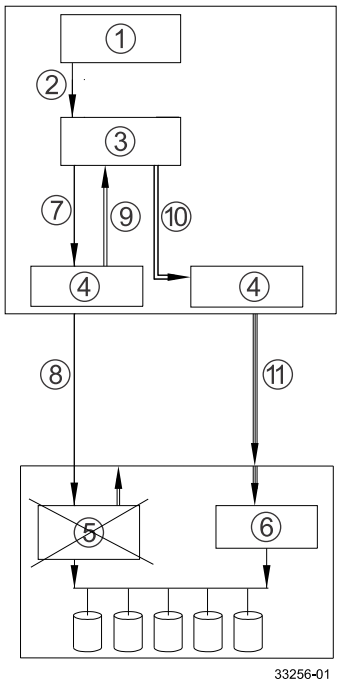
- The multipath driver configuration might require tuning.
- A host system might have a specified volume or volume group reserved, which means that only that host system can perform operations on the reserved volume or volume group.




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

## Supporting redundant controllers

The following figure shows how multipath 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.




1. Host Application
2. I/O Request
3. Multipath 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 Multipath Driver
10. Failover Occurs and I/O Transfers to Another Controller
11. I/O Request Re-sent to Controller B

## How a multipath driver responds to a data path failure

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

- The multipath driver checks for the redundant controller.
- The multipath 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 multipath driver performs a controller failure. The multipath driver provides notification of an error through the OS error log facility.
- The multipath 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)" Enterprise Management Window (EMW) online help topic for the command syntax and description.

# Multipath drivers for the Windows operating system

The multipath 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 array controller 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 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 Read/Write 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 Multipath Device is transferred between TargetPortGroups to one of the following cases:

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

#### Related concepts

*Configurable parameters* on page 21

## 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.

| 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 precedencies are invalidated. |

## ALUA (I/O shipping)

### About this task

I/O Shipping is a feature that sends the Host I/O to a Multipath Device 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 Multipath Device 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.

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

| 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.

| 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.

| 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 can 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.

#### Related concepts

*Configurable parameters* on page 21

## 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. The "Configurable parameters" topic provides a complete list of Wait Times.

#### Related concepts

*Configurable parameters* on page 21

## 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.

**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.

#### Related concepts

*Configurable parameters* on page 21

### 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 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

| Configuration Parameter | Description | Values |
|---|---|---|
| ControllerIoWaitTime | Length of time (in seconds) a request is attempted to a controller before failed over. | Min: 0xA<br>Max: 0x12C<br>Default: 0x78<br>Configured: 0x78 |
| FailedDeviceMaxLogInterval | Specifies the length of time (in seconds) that Test Unit Ready retries will be logged for devices that have not been recovered. | Min: 0x3C<br>Max: 0xFFFFFFFF<br>Default: 0x3C<br>Configured: 0x3C |
| FailedDeviceValidateInterval | Specifies the length of time (in seconds) a Test Unit Ready command is sent to a failed device to determine if it can be recovered. | Min: 0x05<br>Max: 0x3C<br>Max: 0x3C<br>Default: 0xA<br>Configured: 0xA |
| 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<br>Max: 0x3C<br>Default: 0x5<br>Configured: 0x5 |
| 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<br>Max: 0x3C<br>Default: 0x2<br>Configured: 0x2 |

| Configuration Parameter | Description | Values |
|---|---|---|
| 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<br>Max: 0x3C<br>Default: 0x2<br>Configured: 0x2 |

### 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<br>Max: 0x10<br>Default: 0x10<br>Configured: 0x10 |
| NumberOfResidentWorkerThreads | Specifies the number of private worker threads created by the driver. This configuration parameter had been known as NumberOfResidentThreads. | Min: 0x0<br>Max: 0x10<br>Default: 0x10<br>Configured: 0x10 |

### Configurable parameters - path congestion detection

| Configuration Parameter | Description | Values |
|---|---|---|
| CongestionDetectionEnabled | A boolean value that determines whether PCD is enabled. | Min: 0x0 (off)<br>Max: 0x1 (on)<br>Default: 0x0<br>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)<br>Max: 0x1 (yes)<br>Default: 0x0<br>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<br>Max: 0x10000<br>Default: 0x0<br>Configured: 0x0 |
| 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<br>Max: 0x10000<br>Default: 0x0<br>Configured: 0x0 |
| CongestionTimeFrame | A sliding windows that defines the time period that is evaluated in seconds. | Min: 0x1<br>Max: 0x1C20<br>Default: 0x0<br>Configured: 0x0 |

| Configuration Parameter | Description | Values |
|---|---|---|
| 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<br>Max: 0xFFFFFFFF<br>Default: 0x0<br>Configured: 0x0 |
| CongestionMinPopulationSize | The number of sampled I/O requests that must be collected before the average response time is calculated. | Min: 0x0<br>Max: 0xFFFFFFFF<br>Default: 0x0<br>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)<br>Max: 0x1 (yes)<br>Default: 0x0<br>Configured: 0x0 |

### 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<br>Max: 0x1<br>Default: 0x0<br>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<br>Max: 0x3<br>Default: 0x3<br>Configured: 0x3 |
| RetryLunFailoverInterval | Length of time (sec) between additional Failover attempts, if the initial failover request fails. Formally known as "'RetryFailoverDelay". | Min: 0x0<br>Max: 0x3<br>Default: 0x3<br>Configured: 0x3 |
| 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<br>Max: 0x258<br>Default: 0x12C<br>Configured: 0x12C |

| Configuration Parameter | Description | Values |
|---|---|---|
| LunFailoverQuiescenceTime | Length of time (sec) to set in the "QuiescenceTimeout" field of a Legacy Failover request. | Min: 0x1<br>Max: 0x1E<br>Default: 0x5<br>Configured: 0x5 |
| MaxTimeSinceLastModeSense | The maximum amount of time (sec) that cached information regarding TargetPort and TargetPortGroup is allowed to remain stale. | Min: 0x0<br>Max: 0x60<br>Default: 0x5<br>Configured: 0x5 |

### 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<br>Max: 0xFFFFFFFF<br>Default: 0x0<br>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<br>Max: 0xFFFFFFFF<br>Default 0x14<br>Configured: |

### 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<br>Max: 0xFFFF<br>Default: 0x3C<br>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<br>Max: 0xFFFF<br>Default: 0x3C<br>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<br>Max: 0xFFFF<br>Default: 0x41<br>Configured: 0x41 |

### 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)<br>Max: 0x1 (yes)<br>Default: 0x0<br>Configured: 0x0 |

### 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.<br>0x01 - Failover Only<br>0x03 - Round Robin with Subset<br>0x04 - Least Queue Depth<br>0x05 - Least Path Weight<br>0x06 - Least Blocks | Min: 0x1<br>Max: 0x6<br>Default: 0x4<br>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<br>Max: 0xFFFF<br>Default: 0x0<br>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)<br>Max: 0x1 (yes)<br>Default: 0x0<br>Configured: 0x0 |

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

| 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 |

| Event Message | Event Id (Decimal) | Event Severity |
|---|---|---|
| <msg>. Device information is in the DumpData. | 1050 | Informational |

| Event Message | Event Id (Decimal) | Event Severity |
|---|---|---|
| <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 |

| 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 |

| 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 |

| **Event Message** | **Event Id (Decimal)** | **Event Severity** |
|---|---|---|
| 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 |
| 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 |

| Event Message | Event Id (Decimal) | Event Severity |
|---|---|---|
| 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 |
| <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 DumpData. | 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 |

| 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:

| 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 |

| Event Message | Event Id (Decimal) | Severity |
|---|---|---|
| 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 |

| 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 |
| 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 |

| 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 |

| 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 multipath 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] $vsManagementService=gwmi MSVM_VirtualSystemManagementService -namespace "root \virtualization" for each ($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

#### About this task

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.

### Steps

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

Reconfigure the connections between the host and the storage array to remove any redundant connections before you uninstall SANtricity Storage Manager and the DSM multipath driver.

### About this task

**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.

### Steps

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 uses 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

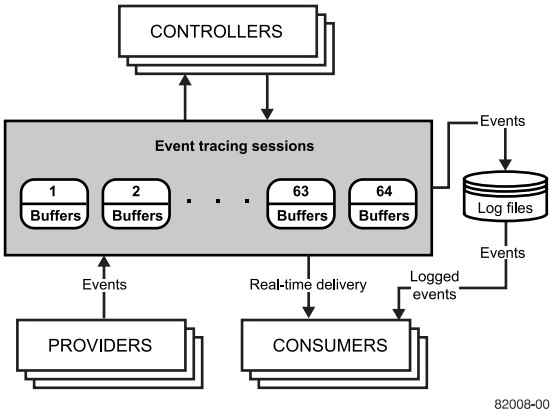
Event Tracing for Windows (ETW) is an efficient kernel-level tracing facility that lets you log kernel or application-defined events to a log file. You can view the events in real time or from a log file and use the events to debug an application or to determine where performance issues are occurring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




### 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 appear 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:

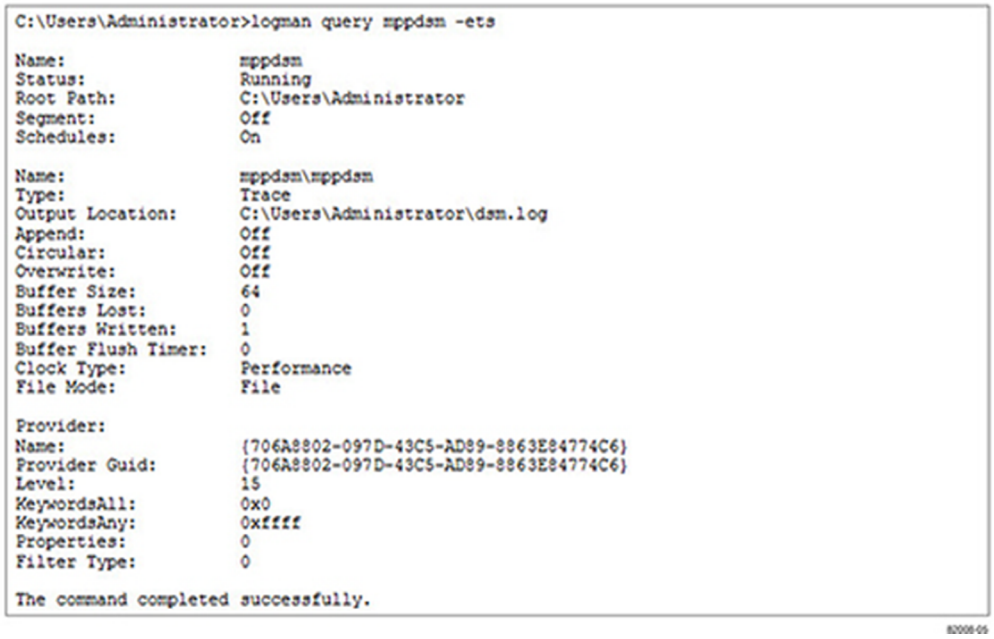
```
C:\Users\Administrator>logman query -ets

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:




### 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 technical support.

## 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.

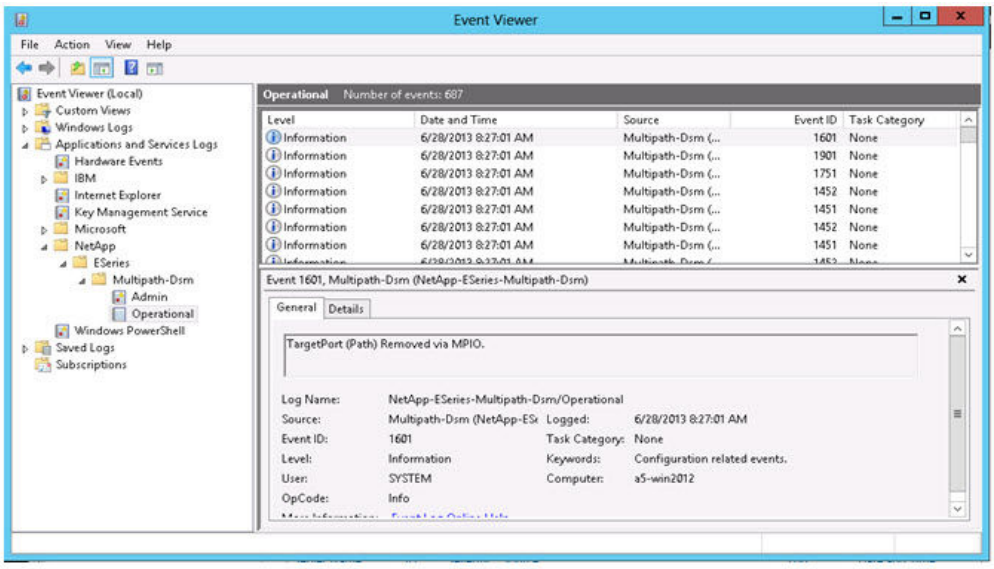
Windows Server 2008 introduced a redesign of the event logging structure that unified the Event Tracing for Windows (ETW) and Event Log APIs. It provides a more robust and powerful mechanism for logging events. Version 2.x of the DSM uses this new approach.

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

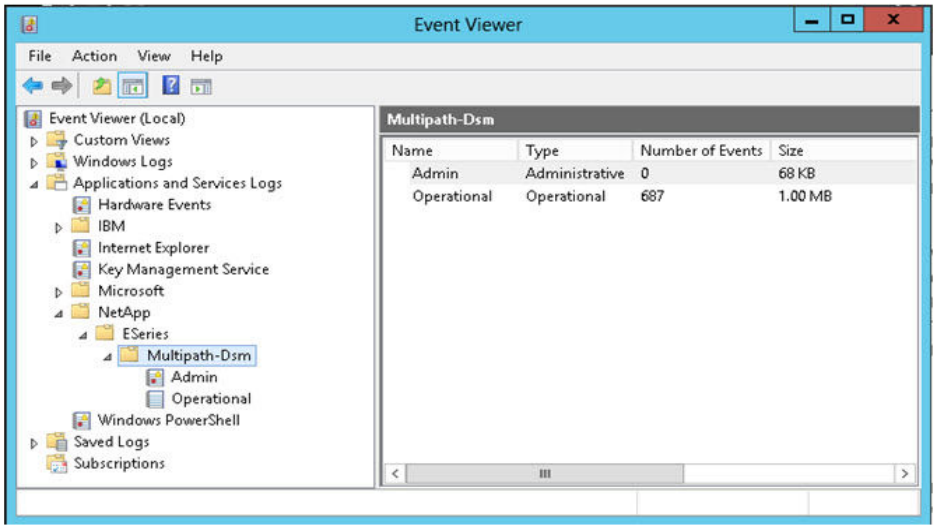
Event channels are viewed using the same Event Viewer application that you use to view the legacy system logs. Currently, the only channel used is the Operational channel.

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.




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.




82008-03

## Loading the custom event view

The following describes a simple procedure for combining both the DSM and the system log information into a convenient single view.

### About this task

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

### Steps

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

Event decoding provides a description of how DSM provides way to store information about an object, and general rules for decoding such information.

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.

# Multipath drivers for the Linux operating system

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

- Device Mapper Multipath (DM-MP) 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 DM-MP is a multipathing driver that is used to communicate with NetApp E-Series and EF-Series storage arrays. It provides an ALUA solution when used with CFW version 7.83 and later.
  - Hosts using this multipath driver should use the Linux (DM-MP) host type in SANtricity.

**Note:** The MPP/RDAC multipath driver is no longer supported on SANtricity 11.25. Refer to the topic on migrating to DM-MP and ensure you are in compliance for failover support by consulting the *NetApp Interoperability Matrix Tool*.

#### Related concepts

*Overview of Migrating to the Linux DM-MP multipath driver* on page 46

## Device mapper multipath (DM-MP) for the Linux operating system

Device Mapper Multipath (DM-MP) 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 - multipath features

DM-Multipath (DM-MP) provides I/O failover and path load sharing for multipathed block devices. DM-MP allows you to configure multiple I/O paths between host and storage controllers into a single device. If one path fails, DM-MP reroutes I/Os to the remaining paths.

**Note:** Co-existence environments (other NetApp products and other vendors) are supported with the DM-MP multipath driver and work independently.

The following list summarizes the features provided by the device mapper:

- 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 multiple paths to maximize performance
- Reconfigures the multiple physical paths to storage automatically when events occur
- Provides DM-MP features support to newly added logical unit
- Provides device name persistence for DM-MP devices under `/dev/mapper/`
- Configures multiple physical paths automatically at an early stage of rebooting to permit the operating system to install and reboot on a multipathed logical unit

## DM-MP load-balancing policies

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

- Round Robin: Loops through every path in the path group, sending the same amount of I/O to each.
- Service Time: 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.
- Queue length: Sends the next group of I/O down the path with the least amount of outstanding I/O.

**Note:** While NetApp makes no recommendations for a path selector to use, both Round Robin and Service Time are appropriate methods. Newer releases default to service-time, which the Linux community believes is the optimal path selector for many situations.

## Known limitations and issues of the device mapper multipath

You should be aware of the known issues, limitations, and workarounds for the device mapper multipath (DM-MP) so that you can use the product more effectively.

The following list summarizes the known limitations and issues associated with the DM-MP multipath driver:

- DM-MP is not capable of detecting changes by itself when the user changes LUN mappings on the target.
  For the appropriate user action, refer to the "Rescanning devices" topic referenced in the "Related concepts" topic immediately after this list.
- DM-MP is not capable of detecting when the LUN capacity changes.
  For the appropriate user action, refer to the "Rescanning devices" topic referenced in the "Related concepts" topic immediately after this list.
- In some operating systems, the DM-MP device partitions are not automatically removed after the partition table is cleared on the device.
  To work around this issue, specify `dmsetup remove <device>`, where `<device>` is the name of the partition device.
- When using "manual" or "followover" failback modes, if the LUN ownership changes by changing the ownership manually, or through a LUN rebalance, the host continues to use the non-optimized path, and the LUN ownership might change to the non-preferred path.
  To work around this issue, run "`multipath -r`" after a manual ownership change, or a LUN ownership redistribution.

### Related concepts

*Rescanning devices with the DM-MP multipath driver* on page 56

## Device mapper operating systems support

For the OS versions supported with the Device Mapper multipath driver and for compatibility information for specific storage array controller firmware versions, refer to the *NetApp Interoperability Matrix Tool*.

## Understanding device handlers

DM-MP 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 section.

- `scsi_dh_rdac`: Plug-in for DM-MP 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 DM-MP 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 plugin, however, is not supported in this release, and is not needed to run ALUA.

## Installing DM-MP

All of the components required for DM-MP are included on the installation media.

### Before you begin

You have installed the following packages on your system.

- For Red Hat (RHEL) hosts, run "`rpm -q device-mapper-multipath`"
- For SLES hosts, run "`rpm -q multipath-tools`"

### About this task

You have installed the following packages on your system.

By default, DM-MP is disabled in RHEL and SLES. Complete the following steps to enable DM-MP components on the host.

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

### Steps

1. Use the procedures in the *Setting up the multipath.conf file* on page 51 to update and configure the `/etc/multipath.conf` file.
2. Enable the `multipathd` daemon on boot.
   - For RHEL 6.x systems, run the following command on the command line: `chkconfig multipathd on`.
   - For SLES 11.x systems run the following commands on the command line: `chkconfig multipathd on` and `chkconfig boot.multipath on`.
   - For RHEL 7.x systems and SLES 12.x, run the following command on the command line: `systemctl enable multipathd`
3. Rebuild the `initramfs` image or the `initrd` image under `/boot directory`.
   a. For SLES 11.x, run the following command on the command line:

      ```
      mkinitrd -k /boot/vmlinux-<flavour> -i /boot/initrd-<flavour>.img -M /boot/System.map-<flavour>
      ```

      In this command, `<flavour>` is replaced with running kernel version from command "`uname -r`".

   b. For RHEL 6.x 7.x and 12.x, run the following command on the command line:

      ```
      dracut --force --add multipath
      ```

c. Make sure that the newly created `/boot/initrams-*` image or `/boot/initrd-*` image is selected in the boot configuration file. For example, for grub it is `/boot/grub/menu.lst` and for grub2 it is `/boot/grub2/menu.cfg` .

4. Do one of the following to verify and, if necessary, change the host type.
   - If you have hosts defined in the **SANtricity Storage Manager Host Mappings View**, go to step 5.
   - If you do not have hosts defined, right-click the default host group in the **SANtricity Storage Manager Host Mappings View**, and then set the default host type to **Linux (DM-MP)**. Go to step 7.
5. In the SANtricity Storage Manager mappings view, right-click the host, and then select **Change Host Operating System.**
6. Verify that the selected host type is **Linux (DM-MP)**. If necessary, change the selected host type to **Linux (DM-MP)**.
7. Reboot the host.

## Overview of Migrating to the Linux DM-MP multipath driver

Because the MPP/RDAC driver is no longer available with SANtricity 11.25, you must migrate to the Linux Device Mapper Multipath (DM-MP) driver.

Migration consists of three steps: preparing for the migration, migrating the MPP/RDAC multipath driver to the Linux DM-MP driver, and verifying the migration to the Linux DM-MP driver.

Preparing for migration is non-disruptive and can be done ahead of time to ensure the system is ready for migration. Migrating to the Linux DM-MP Driver is disruptive because it involves a host reboot.

Downtime for the overall migration procedure involves time taken for the following actions and varies depending on different configurations and running applications:

- Application shutdown procedure
- Host Reboot procedure

### Supported operating systems

Refer to the *NetApp Interoperability Matrix Tool* for supported OS versions for Device Mapper-multipath driver and storage array firmware version. If your operating system and storage array firmware are not in the support matrix for the DM-MP driver, contact technical support.

#### Related tasks

*Preparing to migrate to the DM-MP multipath driver* on page 46
*Migrating the MPP/RDAC driver to the Linux DM-MP driver* on page 47
*Verifying the migration to Linux DM-MP driver* on page 48

## Preparing to migrate to the DM-MP multipath driver

Because the MPP/RDAC driver is no longer available with SANtricity 11.25, you must migrate to the Linux Device Mapper Multipath (DM-MP) driver.

### About this task

The system must be configured to use only persistent device names across all configuration files. This is suggested by all operating system vendors as well. These names are indicated by conventions like */dev/disk/by-uuid* or */dev/disk/by-label* . Persistent names are required because names like `/dev/sda` or `/dev/sdb` might change on the system reboot, depending on the SCSI device

discovery order. Hard coded names can lead to devices disappearing and render the system unable to boot.

To configure persistent device naming conventions in your system, refer to your operating system vendor storage administration guide. NetApp has no recommendation about using specific conventions, provided that the chosen convention is verified by the user and supported by your operating system vendor.

For example, file system table configuration (`/etc/fstab` ) should mount devices and partitions using either, */dev/disk/by-uuid* or */dev/disk/by-label* symbolic names.

#### Steps

1. Mount devices by corresponding /dev/disk/by-uuid names instead of /dev/sd names:

```
UUID=88e584c0-04f4-43d2-ad33-ee9904a0ba32 /iomnt-test1 ext3 defaults
0 2
UUID=2d8e23fb-a330-498a-bae9-5df72e822d38 /iomnt-test2 ext2 defaults
0 2
UUID=43ac76fd-399d-4a40-bc06-9127523f5584 /iomnt-test3 xfs defaults 0
2
```

2. Mount devices by diskname labels:

```
LABEL=db_vol /iomnt-vg1-lvol ext3 defaults 0 2
LABEL=media_vol /iomnt-vg2-lvol xfs defaults 0 2
```

3. Make sure the boot loader configuration file (`/boot/grub/menu.lst for grub` ) uses matching naming conventions. For example, boot loader configurations using filesystem UUID or Label appear as the bold-faced labels in the following two examples:

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

   If you check for SMdevices after migration, you should see one device for each path.

## Migrating the MPP/RDAC driver to the Linux DM-MP driver

Migrating from the MPP/RDAC multipath driver to the Linux DM-MP multipath driver allows you to ensure you have a supported multipath failover solution in SANtricity 11.25.

#### About this task

#### Steps

1. Uninstall the MPP/RDAC driver.

   Typically the default location for the RDAC source directory is under the `/opt/StorageManager/` file path.

   - If the MPP/RDAC driver is installed from the source, go to the RDAC source directory and then run the following command: `#make uninstall`.
   - If the MPP/RDAC driver is installed from RPM, find the linuxrdac package name by specifying `#rpm -q linuxrdac”` and then using the following command to remove it from the system: `#rpm -e “RDAC rpm name"`.

**Note:** Even after uninstalling the MPP/RDAC driver, make sure driver modules (`mppVhba.ko` and `mppUpper.ko` ) remain loaded and running on the system so that application I/O is not disrupted. The host reboot performed in step 4 is necessary to unload these modules.

2. Using a text editor, replace the RDAC-generated initial ram disk image (`/boot/mpp-`uname -r`.img` ) in the boot loader configuration file (for example, `/boot/grub/menu.lst` if using the GRUB boot loader) with the original RAM disk image from when you installed the operating system (that is, `/boot/initrd-<kernel version>.img` or `/boot/initramfs-<kernel version>` file).
3. Install and configure the Linux DM-MP multipath driver.

   Refer to the *Installing DM-MP* on page 45 section to enable and configure the Linux in-box multipath driver. For supported OS versions for DM-MP driver, refer to the *NetApp Interoperability Matrix Tool*.
4. Make sure you properly shut down all your applications.
5. Configure the HBA timeout values for the DM-MP driver, as recommended in the *NetApp Interoperability Matrix Tool* In some cases, DM-MP requires different values than MPP/RDAC, so make sure you verify these settings.
6. Reboot the host.
7. Verify that all file systems are mounted correctly by running the `mount` command.
   - If any of the file systems are not mounted check the `/etc/fstab` file for the corresponding device mount parameters provided.
   - If `/dev/sd` device names are used, change them to either `/dev/disk/by-uuid` symbolic link names or `/dev/mapper/` symbolic names.

## Verifying the migration to Linux DM-MP driver

You can verify that the migration from the MPP/RDAC multipath driver to the Linux DM-MP multipath driver has been successful.

### About this task

After both migration to DM-MP and the host reboot, SMdevices should show multiple entries for the same device, because it should display one device per path.

### Steps

1. Verify that DM-MP device maps are created for all devices with NetApp/LSI vendor ID. Also verify that the path states are `active` and `running` . The priority values for both priority groups of paths should be `14` and `9` respectively as shown in the following example.

   The priority values for both priority groups of paths should be `14` and `9` respectively as shown in the following example. The hardware handler should be rdac and path selector should default as selected by operating system vendors.

   ```
   # 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
```

```
# 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
```

If any of the path states appear as "`ghost`," make sure that Linux(DM-MP) host type is selected from SANtricity Storage Manager Host Mapping view. If any path states appear as "`faulty`" or "`failed`" refer to *Troubleshooting Device Mapper* on page 57. If you require further assistance, contact technical support.

If none of the NetApp/LSI devices appear with these commands, check the `/etc/multipath.conf` file to see if they are blacklisted. If so, remove those blacklisted entries, and then rebuild the initial RAM disk as mentioned in step 2 of *Migrating the MPP/RDAC driver to the Linux DM-MP driver* on page 47.

2. If LVM is configured, run the following commands, and then verify that all the VG/LV/PV devices are referenced by either `WWID` or "`mpath`" names rather than `/dev/sd` 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
```

3. If you encounter any issues, perform the appropriate file system checks on the devices.

## Verifying correct operational mode for ALUA

After setting up the DM-MP multipath driver, you can make that the DM-MP configuration is set up correctly and that ALUA mode is operational.

#### About this task

#### Steps

1. At the command prompt, type `SMdevices`.

   If operating in the correct mode, the output displays either `Active/Optimized` or `Active/Non-optimized` at the end of each line. The host can see the LUNs that are mapped to it.

   If not operating in the correct mode, the output displays either `passive` or `unowned`.

2. At the command prompt, type `multipath -ll`.

   If both controllers are online, there should be exactly two path groups for each LUN, one for each controller, and the number after `prio=` should be greater than 8. The following example displays a device with two paths per controller, for a total of four paths. Each line that starts with `policy=` is a path group. Note the priority values of `14` and `9`:

```
360080e50001b076d0000cd2451ef5e8a dm-8 NETAPP,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='round-robin 0' prio=14 status=enabled
| |- 8:0:2:2  sdn  8:208  active ready  running
| `- 8:0:3:2  sdy  65:128 active ready  running
`-+- policy='round-robin 0' prio=9 status=enabled
  |- 7:0:2:2  sdc  8:32   active ready  running
  `- 7:0:3:2  sdaj 66:48  active ready  running
```

3. Check the syslog messages file, or the output of **dmesg** for the device handler attach message, and ensure that `IOSHIP` appears next to it, in parentheses: `sd 8:0:2:2: rdac: LUN 0 (IOSHIP) (owned)`.

## 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.

The DM-MP multipath driver has built-in default values, as well as built-in settings for different vendor and product ID combinations. When defining sections in multipath.conf, it has the following effects:

- Parameter values defined in the defaults section merge with the built-in defaults, replacing those values.
- Parameter values defined in a device section merge with the built-in defaults for that vendor and product ID if the device already exists in the built-in configuration. To ensure this merging occurs, the vendor and product strings must match the built-in configuration exactly.
- For each parameter, the value is determined in the following sequence:
  1. If defined, the multipath section for each device.
  2. If defined, the device section for the device's vendor and product ID.
  3. The internal default value.

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.

### About this task

The entries should follow the pattern of the following example.

```
blacklist {
       device {
              vendor "*"
              product "Universal Xport"
       }
}
```

## 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, you can update the `/etc/multipath.conf` file . 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 NetApp E-Series and EF-Series devices.

The following example shows part of the `devices` section in the `/etc/multipath.conf` file. 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. NetApp has no recommendation on a particular path selector to use. Therefore, the default path selector will be selected with the device settings as shown in the example. The command "`multipathd show config`" will show the path selector in the defaults 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. For Cluster configurations, set "`failback`" to "`manual`" as specified in the *NetApp Interoperability Matrix Tool*.

```
devices {
  device {
    vendor                      "(LSI|NETAPP)"
    product                     "INF-01-00"
    path_grouping_policy        group_by_prio
    detect_prio                    yes
    prio                        rdac
    path_checker                rdac
    hardware_handler            "1 rdac"
    failback                    immediate
    features                    "2 pg_init_retries 50"
    no_path_retry               30
    retain_attached_hw_handler  yes
         }
}
```

**Note:** The internal default value for path selectors is fine in most cases, and both the round robin or service time path selectors are fully supported.

| Attribute | Parameter Value | Description |
|---|---|---|
| `path_grouping_policy` | `group_by_prio` | The path grouping policy to be applied to this specific vendor and product storage. |
| `detect_prio` | `yes` | The system detects the path policy routine. |
| `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.<br>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.<br>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. |

| Attribute | Parameter Value | Description |
|---|---|---|
| `retain_attached_hw_handler` | `yes` | Specifies that the current hardware handler continues to be used. |

## Setting up DM-MP for large I/O blocks

### About this task

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 (DM-MP) 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 RHEL 6.3, 2048 on SLES 11 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)
  ```

  In this command, `PAGE_SIZE` is architecture independent. It is 4096 for x86_64.

### Steps

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/mpathg` and `/dev/dm-8`. 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.

| 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 as the sysfs, the device mapper, and path checkers |

| Command | Description |
|---|---|
| `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 are accessible through the following path:

```
/dev/mapper/mpathbp1
/dev/mapper/mpathbp2
```

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

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

And the two partitions are accessible through the following path:

```
/dev/mapper/3600a0b80005ab177000017544a8d6b9cp1
/dev/mapper/3600a0b80005ab177000017544a8d6b9cp2
```

## Rescanning devices with the DM-MP multipath driver

In a Linux operating system, you can rescan SCSI devices to work with your new multipath DM-MP driver after installing the new driver.

You can rescan devices through the `rescan-scsi-bus.sh` script, which is included in the **sg3_utils** package.

**Note:** To use this functionality, the operating system must have the **sg3_utils** package version 1.37 or later. To check the package version, specify `rpm -q sg3_utils` from the command line.

| Command | Description |
|---|---|
| `rescan-scsi-bus.sh -m` | Scan for new SCSI devices, and then attempt to create multipath devices. |
| `rescan-scsi-bus.sh -m -u` | Search for remapped SCSI devices, remove the old multipath device, and then create a new multipath device. |
| `rescan-scsi-bus.sh -m -s` | Search for resized SCSI devices, and then update the multipath device size. |

| Command | Description |
|---|---|
| `rescan-scsi-bus.sh -m -r` | Search for unmapped SCSI devices, and then delete both the devices and their associated multipath device. |

## Troubleshooting 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.<br>Verify that the `multipath.conf` file has been updated with proper settings. You can check the running configuration with the `multipathd show config` command. |

# Multipath drivers for the AIX/PowerVM operating system

Multipath I/O (MPIO) is the supported multipath driver for the AIX/ PowerVM operating system on the E-Series or EF-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 multipath drivers, which include:

- MPIO
- SDDPCM
- RDAC

Only the MPIO driver is supported with the E-Series or EF-Series systems.

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 re-routes 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 E-Series or EF-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 E-Series and EF-Series systems.

## Listing the device driver version (MPIO)

#### About this task

**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 E-Series/EF-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 E-Series/EF-Series devices with the AIX/PowerVM operating system.

### Related information

*SANtricity Storage Manager 11.25 Software Installation Reference*
*SANtricity Storage Manager 11.25 System Upgrade Guide*

## 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 E-Series and EF-Series devices.

### About this task

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 SANtricity Storage Manager for AIX.

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

### Step

1. To validate the ODM, run the following command:

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

The expected result displays below:

```
root@ictm-iop-r12-dipper# lslpp -l disk.fcp.netapp_eseries.rte
  Fileset                      Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  disk.fcp.netapp_eseries.rte
                             1.0.600.2  COMMITED   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

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.

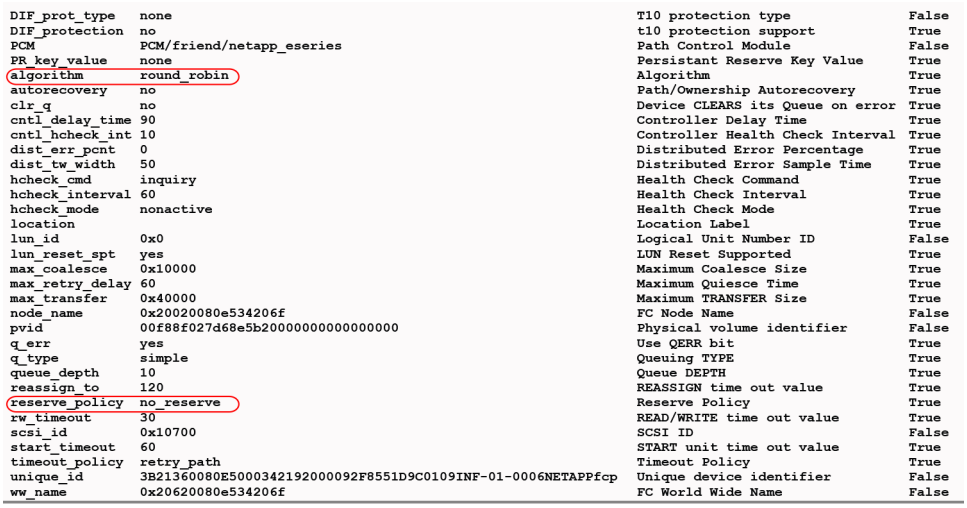
### Checking the AIX default settings

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

```
# lsattr -El hdiskxx
```

In this command, `xx` is the hdisk number.

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



| | | | |
|---|---|---|---|
| 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 | round_robin | 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 | 0x20020080e534206f | FC Node Name | False |
| pvid | 00f88f027d68e5b20000000000000000 | 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 | no_reserve | Reserve Policy | True |
| rw_timeout | 30 | READ/WRITE time out value | True |
| scsi_id | 0x10700 | SCSI ID | False |
| start_timeout | 60 | START unit time out value | True |
| timeout_policy | retry_path | Timeout Policy | True |
| unique_id | 3B21360080E5000342192000092F8551D9C0109INF-01-0006NETAPPfcp | Unique device identifier | False |
| ww_name | 0x20620080e534206f | FC World Wide Name | False |

82013-10

### Checking the HBA settings

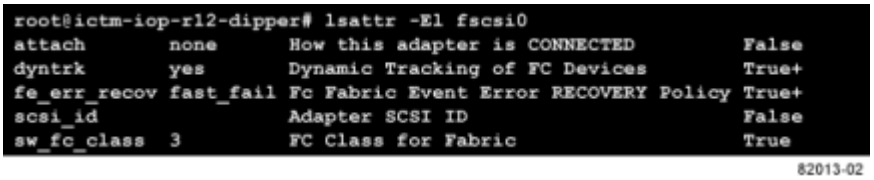
In the latest version of SANtricity firmware (8.25), the following defaults are new.

- `dyntrk=yes`
- `fc_err_recov=fast_fail`

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

```
# lsattr -El fscsixx
```

In this command, `xx` is the fscsi number.



```
root@ictm-iop-r12-dipper# 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

If your system does not display the default settings displayed in the example above, you can 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 failover algorithm

The ODM I/O algorithm is set to round_robin by default. NetApp E-Series and EF-Series systems support the fail_over algorithm as well. You can change the algorithm and the `Reserve_policy` parameter to "`fail_over`" and "`single_path`." This change allows I/O to be routed down one path at a time.

### About this task

Where these commands are run depends on whether you are using the NPIV configuration or the SCSI PowerVM configuration. You have to manually set up the algorithm and the `reserve_policy` to "`fail_over`" and "`single_path`" on E-Series/EF-Series devices.

### Steps

1. To check the default settings for the ODM I/O algorithm, run the following command:

   ```
   # lsattr -El hdiskxx
   ```

   In this command, `xx` is the hdisk number.

2. For each E-Series and EF-Series hdisk in your configuration, run the following `chdev` command to change the algorithm:

   ```
   # chdev -l hdisk1 -a 'algorithm=fail_over reserve_policy=single_path'
   ```

   In this command, `1` is your hdisk number.

   If the algorithm setting was successfully changed, you 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 fails as shown in the following example:

   ```
   E.g (algorithm setting failed) :
   # chdev -l hdisk5 -a 'algorithm=fail_over reserve_policy=single_path'
   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:<br>`# lslpp -l disk.fcp.netapp_eseries.rte`<br>Check the HBA settings and the failover settings.<br>To rescan, enter the following command:<br>`# cfgmgr` |
| Why are no devices listed when I run the`mpio_get_config -Av` command? | This command will not work on AIX/PowerVM with E-Series/EF-Series. Instead, run the following command:<br>`# mpio_get_config -l hdiskxx`<br>In this command, hdiskxx represents the MPIO device on the E-Series/EF-Series storage system. |

# Multipath drivers for the Solaris operating system

MPxIO (using TPGS for Solaris 11) is the supported multipath 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 and 11 OS

MPxIO is included in the Solaris 10 and 11 OS. Therefore, you do not need to install MPxIO . You only need to enable it.

#### About this task

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

#### Steps

1. To enable MPxIO for FC drives, run the following command:

   ```
   stmsboot -D fp -e
   ```

2. Reboot the system.
3. To prepare for either enabling or disabling MPxIO on a specific drive port, specify `ls -l /dev/cfg/` on the command line.

   From the output returned, select the port you would like to either enable or disable.

4. Add the port you want to either enable or disable to the `/kernel/drv/fp.conf` Fibre Channel port driver configuration file by specifying a line similar to the following examples:

**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";
```

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

**Enable**

```
# stmsboot -e
```

**Disable**

```
# stmsboot -d
```

# Editing the sd.conf file and the ssd.conf file for TPGS support in Solaris 10

To ensure the accuracy of the failover process using TPGS support with Solaris 10, you can edit either the sd.conf file or the ssd.conf file.

### About this task

If you update your system frequently with patches as they become available, you do not need to perform this task.

### Steps

1. Depending on your configuration file, edit one of the following configuration files:
   - If your system uses the SPARC architecture, access the `/kernal/drv/ssd.conf` file, and edit the file so it contains the following:

     ```
     ssd-config-list=”NETAPP  INF-01-00 “, “cache-nonvolatile:true,
     disksort:false, physical-block-size:4096,
     retries-busy:30, retries-reset:30, retries-notready:300, retries-
     timeout:10, throttle-max:64, throttle-min:8”;
     ```

     **Note:** There must be two spaces separating the VID (`NETAPP`) from the PID (INF-01-00).

   - If your system uses the x86 architecture, access the `/kernal/drv/sd.conf` file, and edit the file so it contains the following:

     ```
     sd-config-list=”NETAPP  INF-01-00 “, “cache-nonvolatile:true,
     disksort:false, physical-block-size:4096,
     retries-busy:30, retries-reset:30, retries-notready:300, retries-
     timeout:10, throttle-max:64, throttle-min:8”;
     ```

     **Note:** There must be two spaces separating the VID (`NETAPP`) from the PID (INF-01-00).

2. Reboot for these changes to take effect.

## Configuring multipath drivers for the Solaris OS

Use the default settings for all Solaris OS configurations.

## Frequently asked questions about Solaris multipath drivers

Frequently asked questions include questions related to Solaris multipath drivers.

| Question | Answer |
|---|---|
| Where can I find the .conf files that are used by MPxIO? | You can find MPxIO-related files in this directory:<br>`/kernel/drv` |
| Where can I find SANtricity data files? | You can find SANtricity data files in this directory:<br>`/var/opt/SM` |
| Where can I find the command line interface (CLI) files? | You can find CLI files in this directory:<br>`/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:<br>`/dev/rdsk`<br>`/dev/dsk` |
| How can I confirm that MPxIO is enabled? | Check the `format` command output to ensure that the devices have logical paths beginning with `/scsi_vhci/`. |
| Where can I find the SANtricity Storage Manager files? | You can find the SANtricity Storage Manager files in these directories:<br>`/opt/SMgr`<br>`/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.<br>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 either the `SMdevices` utility or the `mpathadm list lu` command to list all volumes and their paths. If you still cannot see any new volumes, reboot the host and then run either the `mpathadm list lu` command again, or the `SMdevices` utility.<br>The `mpathadm list lu` command works only if MPxIO is enabled. As an alternative, list this information by entering either the `luxadm probe` command or the `format` command. |

| Question | Answer |
|---|---|
| How do I find which failover module manages a volume in Solaris? | Check the host log messages (applies to Solaris 11 only) for the volume. Storage arrays with Asymmetric Logical Unit Access (ALUA) are managed by the `f_tpgs` module. Storage arrays with an earlier version of the firmware are managed by the `f_asym_lsi` module.<br>As an alternative, list this information by selecting one of the devices or LUNs you would like to check, and then enter the following command: `# mpathadm show lu <disk>` and check the access state.<br>The system response resembles:<br>`- Active optimzed/Active not optimized for ALUA/TPGS targets`<br>`- Active/Standby for non ALUA/TPGS targets` |
| How can I determine the multipath support for my device? | Use the following command to list the vendors VID.<br>`# mpathadm show mpath-support libmpscsi_vhci.so`.<br>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 after enabling the MPxIO multipath driver? | 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:<br>`fp.conf.enable.20140509_1328` |

# Multipath drivers for the VMware operating system and upgrade instruction

Multipath software within VMware vSphere ESXi is handled natively by the operating system, through NMP (Native Multipathing Plugin). NMP uses various Storage Array Type Plugins (SATPs) to allow for different failover implementations from storage array vendors. E-Series and EF-Series storage arrays use the SATP plugin VMW_SATP_ALUA.

### About this task

SATP rules are inbox for all storage array types (E2700, E5500, E5600, EF550, and EF560) and VMware releases (5.1 U3, 5.5 U2, 5.5 U3, 6.0, and 6.0 U1) supported in the 11.25 SANtricity release. VAAI (vSphere Storage APIs for Array Integration) claim rule modifications are only necessary for ESXi 5.1 releases, as detailed on the *NetApp Interoperability Matrix Tool*.

Starting with ESXi 5.0 U1 and ESXi 4.1 U3, VMware automatically has 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 the TPGS bit disabled are still managed by the VMW_SATP_LSI plug-in.

If upgrading from a VMware release not supported with 11.25 to a VMware release supported with 11.25 on the NetApp Interoperability Matrix Tool, SATP and VAAI claim rule modifications may be required.

### Steps

1. Verify that shell or remote SSH access is available to the ESXi host.
2. Upgrade the controllers in the storage array to controller firmware version 8.25, with the corresponding NVSRAM version.
3. From the host management client, verify that the host OS type is set to *VMWARE*. By default, the *VMWARE* host type enables both ALUA and TPGS.
4. If either of the following conditions applies, specify the code that appears below from the command line:
   - If the VMware host is ESXi 5.5, prior to update release U2 (Build 2068190)
   - If VMware host is ESXi 5.1, including update releases U1, U2, and U3.

```
esxcli storage nmp satp rule add -s VMW_SATP_ALUA -V NETAPP -M
INF-01-00 -c tpgs_on -o reset_on_attempted_reserve -P VMW_PSP_RR

esxcli storage core claimrule remove -r 65433 -c Filter
esxcli storage core claimrule remove -r 65433 -c VAAI
esxcli storage core claimrule add -r 65433 -t vendor -P VAAI_FILTER
-c Filter -V NETAPP -M "LUN*"
esxcli storage core claimrule add -r 65433 -t vendor -P
VMW_VAAIP_NETAPP
-c VAAI -V NETAPP -M "LUN*"
```

# Copyright information

Copyright © 1994–2017 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

# Trademark information

Active IQ, AltaVault, Arch Design, ASUP, AutoSupport, Campaign Express, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Element, Fitness, Flash Accel, Flash Cache, Flash Pool, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, Fueled by SolidFire, GetSuccessful, Helix Design, LockVault, Manage ONTAP, MetroCluster, MultiStore, NetApp, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANscreen, SANshare, SANtricity, SecureShare, Simplicity, Simulate ONTAP, Snap Creator, SnapCenter, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, SolidFire, SolidFire Helix, StorageGRID, SyncMirror, Tech OnTap, Unbound Cloud, and WAFL and other names are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web.

*http://www.netapp.com/us/legal/netapptmlist.aspx*

# How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

*doccomments@netapp.com*

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277