![Sun Microsystems logo]

Sun StorageTek™ MPIO Device Specific Module Installation Guide

For Microsoft Windows OS

Sun Microsystems, Inc.
www.sun.com

Submit comments about this document by clicking the Feedback[+] link at: http://docs.sun.com

Please Recycle

Adobe PostScript™

Contents

Preface

The *Sun StorageTek MPIO Device Specific Module Installation Guide for Microsoft Windows OS* explains how to install the Sun StorageTek MPIO Device Specific Module driver on Microsoft Windows OS platforms. This document is written for technicians, system administrators, and users who have advanced experience with Windows OS.

Related Documentation

The online documentation related to this product is available at the Sun Microsystems Documentation site at:

http://docs.sun.com/app/docs/prod/6180.array#hic

http://docs.sun.com/app/docs/prod/6780.array#hic

http://docs.sun.com/app/docs/prod/6580.array#hic

http://docs.sun.com/app/docs/coll/st2500arrays

http://docs.sun.com/app/docs/coll/st6540array

Documentation, Support, and Training

Sun Function	URL
Documentation	http://www.sun.com/documentation/
Support	http://www.sun.com/support/
Training	http://www.sun.com/training/

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. Submit comments about this document by clicking the Feedback[+] link at http://docs.sun.com. Please include the title and part number of your document with your feedback:

Sun StorageTek MPIO Device Specific Module Installation Guide for Microsoft Windows OS, part number 820-4737-13.

Sun StorageTek MPIO Device Specific Module

The Sun StorageTek Multipath I/O (MPIO) Device Specific Module (DSM) for 6000 and 2500 Series Arrays is available for the following Windows OS versions:

- 32-bit (x86) processors
- 64-bit for the Intel (Itanium or IA64) processor
- 64-bit for AMD Opteron or Intel EM64T (x64) processor

Only one version of the DSM driver can exist on the server because the driver versions are not compatible.

Array Controller Firmware Requirements

You can use MPIO for all controllers that run array controller firmware version 06.19 or later.

Note – The array controller firmware is delivered with the Sun StorageTek Common Array Manager (CAM) software. For the latest patches available for your system, check Sun Solve at: http://www.sunsolve.sun.com.

About MPIO

Microsoft Multipath I/O (MPIO) provides an infrastructure to build highly available solutions for the Windows 2000 operating system (OS), the Windows Server 2003 OS, and the Windows Server 2008 OS. MPIO uses Device Specific Modules (DSMs) to provide I/O routing decisions, error analysis, and failover.

DSM Driver Features

TABLE 1 lists the features provided by the DSM driver.

TABLE 1   Supported MPIO DSM Features

Feature	Description
Native SCSI-2 Release/Reservation (R/R) Commands in a Multipath Environment	If multiple paths exist to a single controller and a SCSI-2 R/R is received for a volume, the DSM driver selects one path to each controller and repeats the request (called a Reservation Path). This function is necessary because the controllers cannot accept SCSI-2 R/R requests through multiple paths for a given volume. After the reservation path has been established, subsequent I/O requests for a volume are restricted to that path until a SCSI-2 release command is received. The DSM driver distributes the reservation paths if multiple volumes are mapped to the host, which distributes the load across multiple paths to the same controller.
Translation of SCSI-2 Reservation/Release Commands to SCSI-3 Persistent Reservations	The DSM driver also supports the ability to translate the SCSI-2 R/R commands into SCSI-3 Persistent Reservations (PR). This allows a volume to use all of the available controller paths rather than being restricted to a single Reservation Path as described above. This feature requires the DSM driver to establish a unique "reservation key" for each host. This key is stored in the Registry and is named "S2toS3Key." If this key is present translations are performed; else, the "cloning" method mentioned above is used.
Windows Failover Cluster	Clustering for Windows Server 2008 uses SCSI-3 Persistent Reservations natively, so the DSM driver does not perform any of the methods listed for SCSI-2 R/R. Translations still occur if the DSM driver is running in a Windows 2000 or 2003 server-based environment.

TABLE 1 Supported MPIO DSM Features *(Continued)*

Feature	Description
Multi-Vendor DSM Support	The MPIO solution relies on the DSM drivers to provide I/O routing, error resolution, and failover resolution. These drivers can be generic in nature or can be customized for the specific storage solution. As a result, multiple DSMs can be installed on a single host, each one potentially managing different storage arrays. Windows allows multiple copies of the same driver to run at the same time; however, the drive's binary name must be different. To avoid conflicts, make sure that only one copy of a failover driver is installed and running.
DSM Driver Naming Conventions	The following naming convention has been established for the DSM driver: Driver names must follow an 8.3 naming convention. This is a restriction of the operating system itself. All DSMs provided by Sun contain the text within the driver name itself, which allows five characters for the OEM identifier.
SynchTimeout Parameter is Evaluated	The SynchTimeout parameter determines the I/O timeout for synchronous requests generated by the DSM driver. Examples include the SCSI-2 to SCSI-3 PR translations and Inquiry commands used during device discovery. It is important that the timeout value for the requests from the DSM driver be at least as large as the timeout value for the MS Disk driver. As a result, the rule above was modified as follows: If the SynchTimeout value is defined in the Registry key of the DSM driver, document it if the TimeOutValue of the MS Disk Driver is defined in the Registry, use the higher of the two values as the SynchTimeout value. If neither value is defined, the value defaults to 10 seconds. For example, if the SynchTimeout is 120 seconds and the TimeOutValue is 60 seconds, use 120 seconds for the value. If the SynchTimeout is 120 seconds and the TimeOutValue is 180 seconds, use 180 seconds for the value of the synchronous I/O requests for the DSM driver.

Supported Storage Arrays

The following storage arrays are supported by the DSM driver.

- Sun Storage 6180 array
- Sun Storage 6580 array
- Sun Storage 6780 array
- Sun StorageTek 6540 array

- Sun StorageTek 6140 array
- Sun StorEdge 6130 array
- Sun StorageTek 2510 array
- Sun StorageTek 2540 array
- Sun StorageTek 2530 array
- StorageTek Flex380 array
- StorageTek B280 array
- StorageTek B210 array
- StorageTek B220 array
- StorageTek B240 array
- StorageTek 9176 array
- StorageTek D173 array
- StorageTek D178 array
- StorageTek D210/Flexline 210 array
- StorageTek D220/Flexline 220 array
- StorageTek D240/Flexline 240 array
- StorageTek D280/Flexline 280 array

Preparing for Installation

For a failover driver to claim ownership over the default `Disk.sys` driver, you must modify two items during installation:

- Modify device node entries in the Registry so that their entries reflect a new class driver.
- When removing a failover driver for any active device, modify the device node entries in the Registry back to their original values.

Installation of the failover drivers takes very little time compared to the required modifications to the device nodes in the Registry. Also, the installation takes very little time compared to restarting the devices.

Set the hardware and compatible IDs so that their IDs return an identifier known only to the failover driver's INF configuration file. Use a filter driver that resides on the HBA bus driver. The HBA bus driver intercepts the identification process.

The plug-and-play (PnP) process is asynchronous, and, because it is event driven, no concept of completion exists. As a result, even though an installation has completed, PnP itself might continue through the driver re-initialization process.

This type of behavior can be due to a target device that has all possible support items entered in the Registry. The DSM driver looking for a device that is not attached, has to wait for an I/O timeout to move to the next device. Also, long delays might occur if you immediately restart a host after failover driver installation, because this process has not yet completed.

Downloading and Installing the MPIO DSM Driver

Multipath DSM drivers are available from the Sun Download Center. Packages are available for the Sun StorageTek 6000 array series (includes Sun Storage 6180, 6580, and 6780 arrays) and Sun StorageTek 2500 array series.

Caution – MPIO and RDAC cannot coexist on the same server.

1. **To download the MPIO driver, go to:**

   http://www.sun.com/download/index.jsp?tab=2#S

2. **Select the package for the appropriate platform version and array family.**

3. **Open the installation program.**

   The installation window appears.

4. **Click Next.**

   The copyright page is displayed.

5. **Click Next.**

6. **Accept the terms of the license agreement, and click Next.**

7. **Click Install.**

8. **Select one of these options, and click Done.**
   - Yes, restart my system
   - No, I will restart my system myself

Windows DSM Configuration Settings

The MPIO failover driver contains configuration settings that can modify the behavior of the driver. Any changes to the settings take effect on the next reboot of the host. The default values listed here are the platform-independent settings. Many of these values are overridden by the failover installer for Windows. For Windows, the configuration settings can be found in the Registry under:

`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\`*DSM_Driver*

where *DSM_Driver* is the name of the OEM-specific driver. The default driver is named `mppdsm.sys`.

Modifying Registry Settings

Caution – Modifying Registry entries should only be done by experienced users.

1. **To view or modify Registry settings, right-click Start and select Run.**

2. **Type `regedit` and click OK.**

Caution – You might lose access to the storage array if you change these settings from their configured values.

TABLE 2 Configuration Settings for Windows DSM

Setting	OS Default Value	Description
MaxPathsPerController	4	The maximum number of paths (logical endpoints) supported per controller. The total number of paths to the storage array is the MaxPathsPerController value multiplied by the number of controllers.
ScanInterval	1	The interval, in minutes, during which the failover driver will check for these conditions: • A change in preferred ownership for a LUN • Attempt to rebalance LUNs to their preferred paths • A change in AVT enabled/disabled status
ErrorLevel	3	Determines which errors to log. The valid range is from 0 to 4. • 0 - Display all errors • 1 - Display path failover errors, controller failover errors, retry errors, fatal errors, and recovered errors • 2 - Display path failover errors, controller failover errors, retry errors, and fatal errors • 3 - Display path failover errors, controller failover errors, and fatal errors (this is the default setting) • 4 - Display controller failover errors and fatal errors
SelectionTimeoutRetry Count	0	The number of times a selection timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If no other path to the same controller exists, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed.
CommandTimeoutRetry Count	1	The number of times a command timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If another path does not exist, a failover takes place. If no valid paths exist to the alternate controller, the I/O fails.

TABLE 2 Configuration Settings for Windows DSM *(Continued)*

Setting	OS Default Value	Description
UaRetryCount	10	The number of times a Unit Attention (UA) status from a LUN is retried. This parameter does not apply to UA conditions due to Quiescence In Progress.
SynchTimeout	120	The timeout, in seconds, for synchronous I/O requests generated internally by the failover driver. Examples of internal requests include those related to rebalancing, path validation, and issuing of failover commands.
DisableLunRebalance	0	Provides control over the LUN failback behavior of rebalancing LUNs to their preferred paths. The following values are possible: • 0 - LUN rebalance is enabled for both AVT and non-AVT modes. • 1 - LUN rebalance is disabled for AVT and enabled for non-AVT mode. • 2 - LUN rebalance is enabled for AVT and disabled for non-AVT mode. • 3 - LUN rebalance is disabled for both AVT and non-AVT modes.
S2ToS3Key	Unique key	Value of the SCSI-3 reservation key generated during failover driver installation.

Wait Time Settings

When the failover driver receives an I/O request for the first time, the failover driver logs timestamp information for the request. If a request returns an error and the failover driver retries the request, the current time is compared with the original timestamp information. Depending on the error and the amount of time that has elapsed, the request is retried to the current owning controller for the LUN or a failover is performed and the request sent to the alternate controller. This process is known as a "Wait Time."

For the Windows OS, the configuration settings can be found in the Registry under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services*<DSM_Driver>*, where *<DSM_Driver>* is the name of the OEM-specific driver. The default driver is named `mppdsm.sys`. Any changes to the settings take effect on the next reboot of the host.

Caution – Possible loss of data access. If you change the Wait Time settings from their configured values, you might lose access to the storage array.

TABLE 3   Wait Time Settings

Configuration Name	OS Default Value	Description
NotReadyWaitTime	300	The time, in seconds, a Not Ready condition (SK 0x06, ASC/ASCQ 0x04/0x01) is allowed before failover is performed.
BusyWaitTime	600	The time, in seconds, a Busy condition is allowed for a failover is performed.
QuiescenceWaitTime	600	The time, in seconds, a Quiescence condition (SK 0x06, ASC/ASCQ 0x8b/0x02) is allowed before a failover is performed.
ControllerIoWaitTime	600	Provides an upper-bound limit, in seconds, an I/O is retried on a controller regardless of retry status before a failover is performed. If the limit is exceeded on the alternate controller the I/O is again attempted on the original controller. This process continues until the ArrayIoWaitTime limit.
ArrayIoWaitTime	600	Provides an upper-bound limit, in seconds, an I/O is retried to the array regardless of which controller the request is attempted to. Once this limit is exceeded the I/O is returned with a failure status.

Path Congestion Detection

The path congestion detection feature allows the DSM driver to place a path offline based on the I/O latency of a path. You can change the settings for this feature by using the dsmUtil utility. Any settings that you change and save with the SaveSettings option will then be found in the Registry under

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*<DSM_Driver>*
where *<DSM_Driver>* is the name of the OEM-specific driver. Any changes to the
settings take effect on the next reboot of the host.

TABLE 1   Path Congestion Detection

Parameter Name	Default Value	Description
CongestionDetection Enabled	0x0	A Boolean value that indicates whether the path congestion detection is enabled. If this parameter is not defined or is set to 0x0, the value is false, the path congestion feature is disabled, and all of the other parameters are ignored. If set to 0x1, the path congestion feature is enabled. The allowed values are 0x0 or 0x1.
CongestionResponse Time	0x0	If CongestionIoCount is 0x0 or not defined, this parameter represents an average response time in seconds allowed for an I/O request. If the value of the CongestionIoCount parameter is nonzero, then this parameter is the absolute time allowed for an I/O request. The allowed values range from 0x1 to 0x10000 (approximately 18 hours).
CongestionIoCount	0x0	The number of I/O requests that have exceeded the value of the CongestionResponseTime parameter within the value of the CongestionTimeFrame parameter. The allowed values range from 0x0 to 0x1000 (approximately 4000 requests).
CongestionSampling Interval	0x0	The number of I/O requests that must be sent to a path before the nth 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. If this parameter is set to 0x0 or not defined, the path congestion feature is disabled for performance reasons every I/O request would incur a calculation. The allowed values range from 0x1 to 0xFFFFFFFF (approximately 4 billion requests).

TABLE 1 Path Congestion Detection *(Continued)*

Parameter Name	Default Value	Description
CongestionMin PopulationSize	0x0	The number of sampled I/O requests that must be collected before the average response time is calculated. The allowed values range from 0x1 to 0xFFFFFFFF (approximately 4 billion requests).
CongestionTakeLast PathOffline	0x0	A Boolean value that indicates whether the DSM driver will take the last path available to the storage array offline if the congestion thresholds have been exceeded. If this parameter is not defined or is set to 0x0, the value is false. The allowed values are 0x0 or 0x1. Note: Setting a path offline with the dsmUtil utility succeeds regardless of the setting of this value.
CongestionTakePaths Offline	0x0	A Boolean value that indicates whether the DSM driver will place a path into an offline state. If this parameter is not defined or set to 0x0, the value is false, and a warning message will be sent to the logs, but the path is not taken offline. If the parameter is set to 0x1, an error message is sent to the logs and the path is taken offline. The allowed values are 0x0 or 0x1. Note: Setting a path offline with the dsmUtil utility will succeed regardless of the setting of this value.

Extended Path Recovery

TABLE 2   Configuration Settings for the Extended Path Recovery Feature

Parameter Name	Default Value	Description
ControllerPath RecoveryTimeFrame	0x3C	The time period during which failed or lost paths can be recovered before a failover is performed to hold the alternate controller in reset. The value is set in seconds. The allowed range of values is 0x3C (60 seconds) to 0x12C (5 minutes).
FailoverEvent ThresholdTimeFrame	0x708	The time period during which the number of failover events are tracked. The value is set in seconds. The allowed range of values is 0x384 (15 minutes) to 0xE10 (60 minutes).
FailoverEvent ThresholdCount	0x03	The maximum number of failover events allowed during the value set in the FailoverEventThresholdTimeFram e parameter. If the number of failover events exceeds the value of this parameter, a failover is performed, holding the failing controller in reset. The allowed range of values is 0x1 to 0x0A.

Configuring Failover Drivers for Windows OS

For the best performance of a redundant controller system, divide I/O activity between the two RAID controllers. You can use the Sun StorageTek Common Array Manager graphical user interface (GUI) or the command line interface (CLI).

To divide I/O activity between two RAID controllers in the management software, perform one of these steps:

- Specify the preferred controller owner when the volume is created, using the Volume Creation wizard.
- Specify the preferred controller owner of a volume at any time, from the Volume Details page in CAM.

You can change the preferred path setting for a volume or a set of volumes online without stopping the applications. The driver uses the new preferred path immediately. Refer to the CAM online help for more details about changing ownership.

An event and an associated alert notification are delivered for a volume that is not on preferred path condition.

Troubleshooting DSM and MPIO Failover Issues

Problem: When installing the DSM driver, you see the message "1633 This installation package is not supported by this processor type."

Solution: The installers are processor dependent. The appropriate file will be of the form: SMIA-*<processor chip>*-*<version>*.exe. The codes are:

WS32 = All 32-bit processors

WS64 = Intel Itanium 64-bit processors only

WSX64 = Intel XEON 64-bit /AMD 64-bit (Opteron) processors

Problem: Disk devices or HBAs show a yellow exclamation point.

Solution: When you use Windows OS Device Manager, you might observe that a disk device or an HBA icon has a yellow exclamation point on it. If new volumes have been mapped to the host, the exclamation point might appear on the icon for a few seconds. This action occurs because the Plug and Play Manager is reconfiguring the device, and, during this time, the device or the HBA might not be used. If the exclamation point stays for more than one minute, a configuration error has occurred.

Problem: Disk devices or HBAs show a red X.

Solution: When you use Windows OS Device Manager, you might notice that a disk device icon or an HBA icon has a red X on it. This X indicates that the device has been disabled. A disabled device cannot be used or communicated with until it is re-enabled. If the disabled device is an adapter, any disk devices that were connected to that adapter are removed from Device Manager.

Problem: Cannot see any volumes.

Solution: If the host has not detected the volumes, an HBA problem or a controller problem has occurred. Make sure that the HBAs are logging into the switch or the controller. If they are not logging in, the problem is probably HBA related.

If the HBAs have logged into the controller, a controller issue might be the problem.

Perform these steps:

1. Make sure that all fiber cables are seated correctly.

2. Make sure that all gigabit interface connectors (GBICs) are seated correctly.

3. Determine the HBA BIOS and driver versions that the system uses, and make sure that the HBA BIOS and driver versions are correct.

4. Make sure that your host-to-volume mappings are correctly mapped in CAM. Do not use any HBA mapping tools.

5. Use WinObj to determine if the host has detected the volumes.

Problem: How do I know if a host has detected my volumes?

Solution: Use WinObj to determine if the host has seen the volumes.

1. If the host does not see the volumes, an HBA problem or a controller problem has occurred.

2. Make sure that the HBAs log into the switch or the controller. If they are not logging correctly, the problem is probably HBA related.

3. If the HBAs have logged into the controller, the problem might be a controller issue.

Problem: When I boot my system, I get a "Registry Corrupted" message.

Solution: Refer to the Microsoft Knowledge Base article 277222 at http:// support.microsoft.com/kb/277222/en-us.

Registry limitations can result in devices and paths that are not recognizable by the host OS and the failover driver.

Problem: My controller failover test does not fail over.

Solution: Make sure that you have looked through the rest of this document for the problem. If you think that the problem is still DSM related, contact a Sun Customer Support representative.

Problem: After I install MPIO, why does my system takes a long time to start?

Solution: You might still experience long start times after you install MPIO because the Windows OS is completing its configuration for each device.

For example, you install MPIO on a host with no storage attached, and you restart the host. Before the Windows OS actually starts, you plug in a cable to a storage array with 32 volumes. In the start-up process, Plug andPlay detects the

configuration change and starts to process it. After the configuration change has completed, subsequent restarts do not experience any delays unless additional configuration changes are detected. The same process can occur even if the host has already started.

Problem: What host type must I use for MPIO/ DSM solutions?

Solution: If you use Microsoft Cluster Server, select a host type of the Windows 2000/Server 2003 clustered OS. If you do not use Microsoft Cluster Server, select a host type of the Windows 2000/Server 2003 non-clustered OS. You set the host type using the Sun StorageTek Common Array Manager (CAM). Select the array you want to configure, go to Administration > Details, and select the appropriate default host type setting.

Note – The Windows 2000/Server 2003 host type settings can also be used for Windows 2008 platforms.

Problem: When I install MPIO on the Windows 2000 OS, the installation fails.

Solution: If you have recently installed Service Pack 4 or Critical Update QFE 813044 to a host, you must install a HotFix (KB822831) to fix a known problem with these updates. After the HotFix has been installed, you can try install or update MPIO again. Determine if Service Pack 4 or the QFE update is installed on your system. Go to the Control Panel on the Start menu, and then select Add/Remove Programs.

Problem: How can I tell if MPIO is installed?

Solution: Perform these steps:

1. Go to the Control Panel on the Start menu, and double-click Administrative Tools.

2. Select Computer Management > Device Manager >SCSI and RAID controllers.

3. Look for Multi-Path Support. If it is present, MPIO is installed.

Problem: How can I tell if the DSM driver is installed?

Solution: Perform these steps:

1. Go to the Control Panel on the Start menu, and then double-click Administrative Tools.

2. Select Computer Management > Device Manager > System Devices.

3. Look for the installed DSM. The name ends with the text "Device-Specific Module for Multi-Path."

If it is present, DSM is installed.

Problem: Why is the path down at the time of registration?

Solution: As part of the SCSI reservation handling, a path might be down at the time that the driver tries to register all paths to a volume. If this is the case, the driver registers all working paths. The failback task performs the registration for the down path after it is working again.

Problem: What should I do if I receive this message? "Warning: Changing the storage array name can cause host applications to lose access to the storage array if the host is running certain path failover drivers. If any of your hosts are running path failover drivers, please update the storage array name in your path failover driver's configuration file before rebooting the host machine to insure uninterrupted access to the storage array. Refer to your path failover driver documentation for more details."

Solution: In the Windows OS, you do not need to update files. The information is dynamically created only when the storage array is found initially. Use one of these two options to correct this behavior:

- Restart the host server.
- Unplug the storage array from the host server, and perform a rescan of all of the devices. After the devices have been removed from the storage array, you can re-attach them. Another rescan takes place, which rebuilds the information with the updated names.

Determining if a Path Failed

With the failover driver, two cases determine if a path has failed:

- A path to a controller has failed, but the driver has other paths to the same controller that it can use. This situation causes a degraded condition and must be corrected before the controller can switch to an alternate path.

  An entry is made in the OS system log that shows that the failover driver has detected a path failure. CAM does not generate an alarm because no internal problems exist for the array.

- All of the paths to a controller have failed, or the controller itself has failed, and the other controller is now used to service the I/O.

  CAM generates the "Volume Not on Preferred Path" alarm for all volumes affected by this scenario. If the array administrator has configured notifications in CAM, the administrator will receive email from CAM or a configured SNMP

server. You also have the option of opening a service request using the Auto Service Request (ASR) feature of CAM. The resultant message and alarm will provide information about the fault, along with possible recovery instructions.

Error Levels

The failover driver has five error levels for messages that are logged to the Windows System Event log:

- Fatal errors
- Controller failover events
- Path failover events
- Retry errors
- Recovered errors

Fatal Errors

TABLE 4 lists the possible Windows OS fatal driver errors.

TABLE 4 Windows OS Fatal Driver Error Messages

Fatal Error Message
Busy wait time exceeded
Busy wait time exceeded on failover command
Close failed on virtual bus node
Command aborted
Command timeout retry count exceeded
Controller unreachable without failback to current (which is disabled)
Discovered device is not a storage array
Duplicate storage array name found
Error trying to allocate data structures
Error trying to create virtual target
Error trying to determine AVT state of volume
Error trying to determine controller slot
Error trying to determine maximum number of volumes supported by storage array

TABLE 4 Windows OS Fatal Driver Error Messages *(Continued)*

Fatal Error Message
Error trying to determine storage array name
Error trying to get World Wide Identifier (WWID) of volume
Error trying to insert a new controller path
Error trying to insert a new volume
Error trying to insert a new volume path
Error trying to match discovered controller to existing controller
Failover command failed
Failover failed, no path to volume
Failover failed, unable to allocate memory
Hardware error
Incorrect close type for virtual bus node
Incorrect open type for virtual bus node
Inquiry wait time exceeded
Maximum number of storage arrays exceeded
Maximum paths per controller exceeded
No paths available to start I/O
No paths available to start I/O, failing over
Not Ready wait time exceeded
Open failed on virtual bus node
Quiescence wait time exceeded
Received the Illegal Command error from the storage array
Request sense failure
Selection timeout retry count exceeded
Sense key hardware error received
Storage array found to be in state of Not Ready and Not Becoming Ready
Unit Attention retry count exceeded
Unrecognizable OS status
Unrecognized SCSI status

TABLE 4   Windows OS Fatal Driver Error Messages *(Continued)*

Fatal Error Message
Unrecognizable sense key received
Volume number exceeds the maximum configured volume number
World Wide Identifier (WWID) for discovered volume does not match that of same volume found on another path

Controller and Path Failover Events

The following items are examples of failover driver controller events and path failover events:

- Change of AVT setting detected
- Failover command issued
- AVT failover invoked
- Path failure detected

WinObj

You can use WinObj to view the Object Manager namespace that is maintained by the operating system. Every Windows OS driver that creates objects in the system can associate a name with the object that can be viewed from WinObj. With WinObj, you can view the volumes and paths that the HBAs have identified. You can also view what the failover driver identifies from a storage array.

For more information about WinObj, see:
http://www.microsoft.com/technet/sysinternals/SystemInformation/WinObj.mspx

DSM

The directory structures for the DSM driver include these paths:

- \Device\MPPDSM - This structure contains information that is maintained by the DSM driver.

- `\Device\Scsi` - This structure contains information that is maintained by the ScsiPort driver.

TABLE 5 Object Path and Descriptions of the WinObj DSM

Object Path	Description
\Device\MPPDSM	The root directory for all named objects that are created by the DSM driver.
\Device\MPPDSM *<storage array>*	The root directory for all named objects that are created by the storage array named *<storage array>*.
\Device\MPPDSM*<storage array>**<ctlr>*	The root directory for all named objects that are created for a given controller. The *<ctlr>* value can either be A or B.
\Device\MPPDSM*<storage array>**<ctlr>*\P*<port>*P*<path>*I*<id>*	The root directory for all named objects that are created for a given path to a controller. The *<port>* value, the *<path>* value, and the *<id>* value represent a SCSI address from a given HBA port.
\Device\MPPDSM*<storage array>**<ctlr>*\P*<port>*P*<path>*I*<id>**<lun>*	The *<lun>* value represents the volume number assigned to the device for a given controller/path combination.
\Device\Scsi	The root directory for all of the named objects that are created by the ScsiPort driver. Each object represents a physical path found by a given HBA.
\Device\Scsi\<adapter>Port<port>\Path<path>Target<target>Lun<lun>	ScsiPort-based HBA drivers A named device object that represents a drive. The *<adapter>* value represents the HBA vendor. For QLogic, this value is based on the HBA model number (for example, ql2300). The *<port>* value, the *<path>* value, and the *<target>* value represent the location of the volume on the HBA.
\Device*<auto-generated id>*	StorPort-based HBA driver. An auto-generated named device object representing a drive.

With this information, you can reach these conclusions:

- The objects shown in the \Device\Scsi directory show the physical volumes identified by the HBAs. If a specific volume is not in this list, the DSM driver cannot detect the volumes.

- The objects shown in the `\Device\MPPDSM` directory show the items that are reported by MPIO to the DSM driver. If a device is not in this list, MPIO has not notified the DSM driver.

Viewing Driver Information in Device Manager

Device Manager is part of the Windows operating system. To view information about the driver:

1. **Select Control Panel from the Start menu.**

2. **Select Administrative Tools > Computer Management > Device Manager.**

   The Device Manager tree for MPIO bases the volume names on the vendor information and product ID information of the underlying physical device, along with the text Multi-Path Disk Device.

3. **Scroll down to System Devices to view information about the DSM driver itself.**

   The Disk Drives section shows both the drives identified by the HBA drivers and the volumes created by MPIO.

4. **Select one of the MPIO volumes, and right-click it.**

5. **Select Properties to show the Multi-Path Disk Device Properties window.**

   This properties window shows if the device is working correctly.

6. **Select the Driver tab to view the driver information.**

Multipath Failover Driver Utility

The `dsmUtil` utility is a general purpose command-line driven utility that works only with MPIO-based DSM solutions. The utility is used primarily as a way to instruct the DSM driver to perform various maintenance tasks, but the utility can also serve as a troubleshooting tool when necessary.

To use the `dsmUtil` utility, type this command, and press Enter.

`dsmUtil` [[-a [*target_id*]] [-c *array_name* | `missing`] [-d *debug_level*] [-e *error_level*] [-g *virtual_target_id*] [-o *feature_action_name*[=*value*][, `SaveSettings`]] [-M] [-P [`GetMpioParameters` | `MpioParameter`=*value* | ...]] [-R] [-s "failback" | "avt" | "busscan" | "forcerebalance"] [-w *target_wwn*, *controller_index*]

Note – The quotation marks must surround the parameters.

Typing dsmUtil without any parameters will show the usage information..

TABLE 6   dsmUtil Parameters

Parameter	Description
–a [*target_id*]	Shows a summary of all storage arrays seen by the DSM driver. The summary displays the target_id, storage array WWID, and storage array name. If target_id is specified, DSM point-in-time state information appears for the storage array. On Unix OSes, the virtual HBA specifies unique target IDs for each storage array. The Windows MPIO virtual HBA driver does not use target IDs. The parameter for this option can be viewed as an offset into the DSM driver information structures, with each offset representing a different storage array.
–c *array_name*| missing	Clears the WWN file entries. This file is located in the Program Files\DSMDrivers\mppdsm\WWN_FILES directory with the extension .wwn. If the *array_name* keyword is specified, the WWN file for the specific storage array is deleted. If the missing keyword is used, all WWN files for previously attached storage arrays are deleted. If neither keyword is used, all of the WWN files, for both currently attached and previously attached storage arrays are deleted.
–d *debug_level*	Sets the current debug reporting level. This option only works if the RDAC driver has been compiled with debugging enabled. Debug reporting is comprised of two segments. The first segment refers to a specific area of functionality, and the second segment refers to the level of reporting within that area. The *debug_level* is one of these hexadecimal numbers: • 0x20000000—Shows messages from the RDAC driver's init() routine. • 0x10000000—Shows messages from the RDAC driver's attach() routine. • 0x08000000—Shows messages from the RDAC driver's ioctl() routine. • 0x04000000—Shows messages from the RDAC driver's open() routine. • 0x02000000—Shows messages from the RDAC driver's read() routine. • 0x01000000—Shows messages related to HBA commands. • 0x00800000—Shows messages related to aborted commands. • 0x00400000—Shows messages related to panic dumps. • 0x00200000—Shows messages related to synchronous I/O activity. • 0x00000001—Debug level 1. • 0x00000002—Debug level 2. • 0x00000004—Debug level 3. • 0x00000008—Debug level 4. These options can be combined with the logical or operator to provide multiple areas and levels of reporting as needed.

TABLE 6 `dsmUtil` Parameters *(Continued)*

Parameter	Description
-e *error_level*	Sets the current error reporting level to error_level, which can have one of these values: 0—Show all errors. 1—Show path failover, controller failover, retryable, fatal, and recovered errors. 2—Show path failover, controller failover, retryable, and fatal errors. 3—Show path failover, controller failover, and fatal errors. This is the default setting. 4—Show controller failover and fatal errors. 5—Show fatal errors
-g *target_id*	Displays detailed information about the state of each controller, path, and LUNs for for the specified storage array. You can find the *target_id*, by running the `dsmUtil -a` command.
-M	Shows the MPIO disk-to-drive mappings for the DSM driver.

TABLE 6   `dsmUtil` Parameters *(Continued)*

Parameter	Description

`-o`
[*feature_action_name=value*]
[`,SaveSettings`]
	Troubleshoots a feature or changes a configuration setting. Without the `SaveSettings` keyword, the changes only affect the in-memory state of the variable. The `SaveSettings` keyword changes both the inmemory state and the persistent state. Some example commands are:
	
	`dsmUtil -o`—Displays all the available feature action names.
	
	`dsmUtil -o DisableLunRebalance=0x3`—Turns off the DSMinitiated storage array LUN rebalance (affects only the in-memory state).
	
	Extended Path Recovery Options—For a description of this feature, see "Windows DSM Configuration Settings" on page 6.
	
	`ControllerPathRecoveryTimeFrame=seconds`
	
	`FailoverEventThresholdTimeFrame=seconds`
	
	`FailoverEventThresholdCount=number_of_events`
	
	An example command is:
	
	`dsmUtil -o ControllerPathRecoveryTimeFrame=70, SaveSettings`—Sets the time during which the path can be recovered before a failover occurs to 70 seconds. Saves the setting in the Registry so it persists after host reboots.
	
	Path Congestion Detection Options—For a description of this feature, see "Windows DSM Configuration Settings" on page 6.
	
	`CongestionDetectionEnabled=value`
	
	`CongestionResponseTime=seconds`
	
	`CongestionIoCount=num_IO_requests`
	
	`CongestionTimeFrame=seconds`
	
	`CongestionSamplingInterval=num_IO_requests`
	
	`CongestionMinPopulationSize=num_IO_requests`
	
	`CongestionTakeLastPathOffline=value`
	
	`CongestionTakePathsOffline=value`
	
	`SetPathOffline=path_ID`
	
	`SetPathOnline=path_ID`
	
	Example commands are:
	
	`dsmUtil -o CongestionDetectionEnabled=0x1`—Enables the path congestion feature. Before it can be enabled, the CongestionResponseTime parameter, the CongestionTimeFrame parameter, and the CongestionSamplingInterval parameter must be set to valid values.
	
	`dsmUtil -o SetPathOffline=0x77070001`—Manually sets a path to offline. Use the PathID value from the command `dsmUtil -g` as the value.

TABLE 6 `dsmUtil` Parameters *(Continued)*

Parameter	Description
`-P [GetMpioParamet ers	MpioParameter= value	...]`	Displays and sets MPIO parameters.
`-R`	Remove the load balancing policy settings for inactive devices.
`-s ["failback"	"avt"	"busscan"	"forcerebalance "]`	Manually initiates one of the DSM driver's scan tasks. A `failback` scan will cause the DSM driver to reattempt communications with any failed controllers. An `avt` scan causes the DSM driver to check whether AVT has been enabled/disabled for an entire storage array. A `busscan` scan causes the DSM driver to go through its unconfigured devices list to see if any of them have become configured. A `forcerebalance` scan causes the DSM driver to move storage array volumes to its preferred controller and ignores the value of the DisableLunRebalance configuration parameter of the DSM driver.
`-w` *target_wwn*, *controller_index*	Sets the World wide identifier for the controll.er