

# **MPT Windows Driver Release Notes for LSI\_SCSI.SYS, LSI\_FC.SYS, LSI\_SAS.SYS (*StorPort Versions*)**

**XP x64, Server 2003, Server 2008, Vista,  
Windows 7 / Server 2008 R2**

**Version 1.33.01**

**04/26/2010**

## **Compatibility:**

**LSI\_SCSI:** LSI53C1020 / LSI53C1020A / LSI53C1030

**LSI\_FC:** LSIFC919 / LSIFC919X / LSIFC929 / LSIFC929X / LSIFC949X  
LSIFC949E

**LSI\_SAS:** LSISAS1064 / LSISAS1068 / LSISAS1078 / LSISAS1064E  
LSISAS1068E

## **Special Notes for 1.33.01:**

- Starting in release 1.32.02, for SAS adapters only, SATA devices will have a default queue depth of 32 vs. 64 for all SCSI, Fibre Channel, and SAS devices. A new registry entry of "MaxSATAQueueDepth" is available to change the default SATA queue depth. Starting with release 1.33.01, a new registry entry of "MaxSASQueueDepth" is available to change the default SAS queue depth of 64.
- Port I/O usage has been added back into the driver for parallel SCSI only. (due to H/W errata). Port I/O is not used for Fibre Channel or SAS.

- Starting in release 1.30.03, the MaxSGList registry entry has been added to allow the user to set the number of SG entries supported by the driver (default 0x101).
- In release 1.30.03, the following four PnP IDs have been added to the Isinodrv.inf file.
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS3\_\_\_3GSAS
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS4\_\_\_3GSAS
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS5\_\_\_3GSAS
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS6\_\_\_3GSAS
- The Windows WDK has a static code analysis tool called Prefast. It scans the code during the driver build process and lists “defects” (problems that are common in drivers). In most cases, the defect is a detection of usage of an uninitialized variable. However, Prefast will issue a defect even if there is no real code path that results in an uninitialized variable. Almost all of the Prefast defects can be resolved by cosmetic changes to the code which do not change functionality. These defects have been fixed in version 1.29.01.
- Starting with 1.27.01, support has been removed for Vista on the IA64 platform.
- Starting with 1.26.03, support was added for Server 2008 production build.
- Driver description strings use the string ‘LSI’ instead of ‘LSI Logic’ starting with version 1.25.08. All user documentation has been changed to reflect the new name.
- Starting with version 1.25.07, the Windows driver does not require PCI Port I/O resources. However, the elimination of Port I/O resources is controlled by a registry entry that is not part of the standard driver package. This is not included as it can cause bluescreens if a driver is downgraded to an earlier version which does require Port I/O resources. A special INF file that sets the NoPortIoUsage registry entry can be requested from LSI.

- Driver installation packages have been restructured, with each package supporting a particular OS version/platform combination.
- Drivers for Microsoft Vista are now available starting with version 1.25.02. These drivers are Release Signed by LSI, allowing for fresh installation and driver upgrades without special settings or attaching a debugger. Release signing is NOT WHQL Logo signed.
- Versions of LSI\_SCSI, LSI\_SAS, and LSI\_FC at or before 1.24.03 can cause memory corruption under a very specific system environment. See the “Defects fixes” section under version 1.24.04
- The BIOS Config Utility settings for sync, wide, and adapter SCSI ID are the only ones honored by the Windows driver. Other settings are overridden by the Windows OS.
- For full driver upgrade functionality in flashless environments, 1030 F/W at revision 1.00.14.00 or above should be used.
- The Server 2003 RTM IA64 version of the StorPort port driver has severe bugs that do not allow the LSIMPT driver to function properly. Attempting to use this version of StorPort on an IA64 platform will crash the system. Apply the KB823728 StorPort hotfix or the latest Server 2003 Service Pack.
- The Server 2003 RTM i386 version of the StorPort port driver has a bug in a timer routine. In a system with a 1030 Intergrated RAID volume, if a RAID physical drive is hot swapped, the newly arrived drive will remain at async/narrow, which will impact performance. To resolve this issue, apply the KB823728 StorPort hotfix or the latest Server 2003 Service Pack.
- When a non-Microsoft MPIO multi-path filter driver is installed, the “MPIOMode=1;” registry entry should be used. When the Microsoft MPIO multi-path driver is installed, the “MPIOMode=2;” registry entry should be used.
- Starting with version 1.20.12, the LSIMPT driver has been split into 3 drivers by underlying protocol: LSI\_SCSI, LSI\_FC, and LSI\_SAS. This was required by Microsoft to support differences in port driver behavior controlled by the BusType registry entry, which is set in the Services\<driver\_name> key.

- For versions 1.20.06 thru 1.20.11, the LSIMPT driver does not support parallel SCSI (U320) devices. This was due to numerous problems with the StorPort driver supporting non-disk devices in a parallel SCSI environment. Removing this support eliminated the problem of U320 devices being switched to the StorPort driver when a SAS or Fibre Channel devices is installed. However, Microsoft has committed to support U320 devices on StorPort. Parallel SCSI support was reinstated with the driver split activity of version 1.20.12.
- To enable End-to-End Data Protection (EEDP) support for the 949X/939X devices, the registry entry "EEDP\_T10\_Enable=1;" must be added to the DriverParameter string.
- For Fibre Channel adapters, the default setting for host driver persistency table support is now on. To turn this off, place the string LUN\_Pers=0; in the DriverParameter string in the registry.

#### **Current Errata:**

- None

# Version 1.33.01

## Major Changes From Version 1.32.02:

### General Changes

#### Functionality

- For SAS adapters only – SATA devices will have a default queue depth of 32 vs. 64 for SAS devices. A new registry entry of MaxSASQueueDepth is available to change this SAS default.
- Added SAS2 Switch (Lynx 6160) to No Driver INF files.

#### Defect fixes

- Fixed wrong index being used for SMFATrackingArray in CompleteAllSrb routine. Index should start at 0 and not 1. Could result in a bluescreen when a diag reset is performed.
- When accessing config pages during driver initialization, increased timeout from 2 to 10 seconds.

#### Enhancements

- None.

# Version 1.32.02

## Major Changes From Version 1.32.01:

### General Changes

#### Functionality

- For SAS adapters only – SATA devices will have a default queue depth of 32 vs. 64 for SAS devices. A new registry entry of MaxSATAQueueDepth is available to change this default. (Performance improvement for SATA devices as most SATA devices have a max queue depth of 32.)

#### Defect fixes

- Fixed a possible bluescreen issue if a diag reset IOCTL timed out. The IOCTL would be completed in the IOCTL timeout routine and would attempt to be completed again in the diag reset recovery routine.

#### Enhancements

- None.

# Version 1.32.01

## Major Changes From Version 1.31.02:

### General Changes

#### Functionality

- For parallel SCSI only – Port I/O usage was reinstated for F/W download boot capabilities. U320 H/W errata does not allow for memory mapped support of F/W download boot on all systems.
- Added PciIoSpaceNotRequired to Fibre Channel and SAS INF files (not parallel SCSI as it needs Port I/O for F/W download boot).

#### Defect fixes

- Fixed lost I/O problem (starting an I/O via tracking array while in ISR thread). Changed WMI SendHbaApiConfigMessage to reuse tracking buffer on ISR calls (rather than use a new buffer).
- Fixed issue with double-completion of an IOCTL when the IOCTL times out (seen with IOCTL activity during tape drive operations).

#### Enhancements

- None.

# Version 1.31.02

## Major Changes From Version 1.30.01:

### General Changes

#### Functionality

- None.

#### Defect fixes

- None.

#### Enhancements

- Integrated Windows Driver Configuration Utility (WDCFG) into driver.
- Added registry entry of "Enable520B=1:" to enable this special 520-byte sector proprietary CRC handling. Utilized the SCSI IO 32 EEDP message to allow reading of a 520-byte sector but only DMA the 512 bytes expected by Windows. For this special 520-byte processing, the EEDP operation is only to strip the DIF information from the data stream (the last 8 bytes) and no checks are performed on any of the DIF fields.



# Version 1.31.01

## Major Changes From Version 1.30.04:

### General Changes

#### Functionality

- Added support for Windows 7 / Server 2008 R2 operating systems.

#### Defect fixes

- None.

#### Enhancements

- Special functionality was added in support of a proprietary OEM utility for FC drives that are formatted to 520-byte sectors (proprietary CRC mechanism). This functionality is enabled via a registry entry and it allows Windows to recognize the 520-byte drives during enumeration.

# Version 1.30.04

## Major Changes From Version 1.30.03:

### General Changes

#### Functionality

- None.

#### Defect fixes

- None.

#### Enhancements

- None.

#### Notes

- The driver version was incremented for the Phase 17 general release. This general release for Phase 17 captures all changes to the driver since version 1.29.03.

# Version 1.30.03

## Major Changes From Version 1.30.02:

### General Changes

#### Functionality

- None.

#### Defect fixes

- None.

#### Enhancements

- Starting in release 1.30.03, the MaxSGList registry entry has been added to allow the user to set the number of SG entries supported by the driver (default 0x101).
- In release 1.30.03, the following four PnP IDs have been added to the Isinodrv.inf file.
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS3\_\_\_3GSAS
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS4\_\_\_3GSAS
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS5\_\_\_3GSAS
  - %SAS\_BCS.DeviceDesc% = NODRV, SCSI\EnclosureIBM-ESXSBCS-DSS6\_\_\_3GSAS

# Version 1.30.02

## Major Changes From Version 1.30.01:

### General Changes

#### Functionality

- None.

#### Defect fixes

- System can bluescreen if connected to a target that supports Broadcast Async Event notification and a task management reset is issued to the target.
  - Seen by: LSI test lab
  - To reproduce: Connect a SAS target that supports Broadcast Async Event notification. Issue I/O's and periodically issue task management resets. If the system bluescreens the problem is reproduced.
  - Root cause: When a Broadcast Async Event (BAE) notification is received the driver uses special MPI message frames to issue config page and task management requests. These frames have a special cotext value (BC\_AEN\_CONTEXT) so the replies can be trapped by the ISR and routed to the BAE routine. A coding error was allowing some of these special replies to fall into code that used the context value as an index into the IO tracking array.
  - Fix: Added a specific check for BC\_AEN\_CONTEXT early in the ISR to properly route these replies to the BAE routine.

# Version 1.30.01

## Major Changes From Version 1.29.03:

### General Changes

#### Functionality

- Added LSI A.1 Expander NoDriver entries to LSINODRV.INF files to suppress asking for drivers when these enclosures are present.
- Added custom OEM tape drive in CD-ROM mode detection for In-Line Data Padding (IDP) (parallel SCSI only).

#### Defect fixes

- SRB\_FUNCTION\_RESET\_BUS was not completing the SRB properly.
  - Seen by OEM customer.
  - To reproduce: Using a kernel mode service, issue an SRB with SRB\_FUNCTION\_RESET\_BUS. The SRB will not be properly completed back to the service. The SRB will have an SRB status of SRB\_STATUS\_PENDING.
  - Fix: Add code to properly complete the SRB back to the port driver.

# Version 1.29.03

## Major Changes From Version 1.29.02:

### General Changes

#### Functionality

- Added two more OEM SAS Enclosure NoDriver entries to LSINODRV.INF files to suppress asking for drivers when these enclosures are present.

#### Defect fixes

None.

# Version 1.29.02

## Major Changes From Version 1.29.01:

### General Changes

#### Functionality

- Added more OEM SAS Enclosure NoDriver entries to LSINODRV.INF files to suppress asking for drivers when these enclosures are present.

#### Defect fixes

- Backed out the change for the problem in the way the driver calculates Global Credits available in the controller.
  - Seen by internal testing.
  - Root cause: The change to count each buffer used actually makes the problem of running out of resources worse, because we are forced to call StorPortBusy more often than the old way, and the change doesn't solve the problem it was intended to solve (avoid reset failures due to I/O's stuck in the request buffer not being terminated by the reset).
  - Fix: Backout this change made in v1.29.01.

# Version 1.29.01

## Major Changes From Version 1.28.03:

### General Changes

#### Functionality

- Removed unused header files from being included in project.

#### Defect fixes

- Aborted I/O is not being retried and returns GOOD status and causes data corruption.
  - Seen by OEM.
  - Root cause: Driver was not checking the status of an aborted I/O resulting in GOOD status being returned and therefore the I/O was not retried. Some new controllers return the SCSISTAT\_TASK\_ABORTED status, but this is not checked and a GOOD status is returned.
    - Fix: Check status of aborted I/O to see if it is SCSISTAT\_TASK\_ABORTED. If so, return SRB\_STATUS\_ERROR instead of SCSISTAT\_GOOD.
- Config Page replies can be processed as Broadcast Async Event message replies.
  - Seen by code review.
  - Root cause: Driver does not separate processing of BAE replies from config page replies.
    - Fix: Isolate processing of BAE messages from config page and TM reset requests by allocating a dedicated static message frame for BAE processing only.
- LUN/Target reset commands left pending in ResetSrbQueue.
  - Seen by internal testing.
  - Root cause: It was observed that LUN/target resets were being left pending in the ResetSrbQueue and were not being processed. This could happen if a diagnostic reset had just completed, and a bus reset (issued from the Hazard test program) was being processed when



Storport subsequently issued LUN and target resets. Those resets were placed in the ResetSrbQueue, but the queue was not checked after the bus reset had completed.

- Fix: Add a check of the ResetSrbQueue for any pending LUN/target resets after any Task Management command is completed.

- Remove redundant StorPortPause calls.
  - Seen by code review.
  - Root cause: It was discovered that if a bus reset fails due to I/O's still outstanding then a StorPortPause call was being made. The I/O queue had already been paused at the start of the bus reset processing, so this second call was redundant. Storport should save us from this, but the redundant call should be remove.
  - Fix: Remove redundant StorPortPause calls if bus reset fails.
- LUN/Target reset may be completed twice, causing bluescreen.
  - Seen by OEM/Internal testing.
  - Root cause: This can occur when Storport issues a LUN or target reset just as a diagnostic reset is being processed. The diagnostic reset calls StorPortCompleteRequest to ensure that all outstanding I/O's have been completed. But, that can also mark the LUN or target reset in flight as being completed, although the port driver will still issue that reset to the miniport. If the miniport does not detect that the reset has already been completed and tries to complete it again the system will bluescreen.
  - Fix: This issue had been seen before and code was added previously to try to detect when a reset has already been completed. This fix adds additional checks of the reset on initial receipt from the port driver, when the reset is pulled from the ResetSrbQueue, and when the reset reply is received from the F/W.
- “Surprise Removal” of adapter can cause bluescreen.
  - Seen by OEM.
  - Root cause: If an adapter is “surprise removed” then the driver must detect this and inform the OS that the adapter is missing. If the driver has detected the “adapter gone” condition, but a new I/O is received

before the OS has processed the adapter gone status, the system can bluescreen due to our StartIo routine attempting to access our H/W.

- Fix: Set an adapter gone flag when the adapter is detected as missing, and check this flag on each new I/O. If the adapter is gone do not attempt to start the I/O but return SRB\_STATUS\_NO\_HBA.

- Problem in the way the driver calculates Global Credits available in the controller.

- Seen by internal testing.
- Root cause: The F/W Global Credits value is the number of message frames allocated by F/W to process I/O's. However, when processing large I/O's with many chain buffers, the F/W will pull all of the chains into controller memory at the start of I/O processing, using additional message frames. The host driver was using Global Credits as the number of concurrent I/O's it can issue to the F/W. With a heavy I/O load of large I/Os many I/O's will be backed up in the H/W Request FIFO. I/O's with shorter timeouts (such as bus enumeration I/O's) may time out due to this backlog. Task Management commands issued to clean up the I/O may not return the timed out I/O if it is still pending in the Request FIFO.

- Fix: Modify the driver to count the number of message frames and chain buffers as a running count of the total number of buffers used, and compare that to the F/W Global Credits. When the value exceeds the Global Credits the driver will stop the new I/O queue until some of the outstanding I/O's complete (reducing the number of outstanding buffers used).

# Version 1.28.03

## Major Changes From Version 1.28.02:

### General Changes

#### Functionality

- Started logging internal device reset completion event if LogExceptionEvents is turned on.
- Added registry method for entering the four dwords of Product Specific data needed for immediate diagnostic buffers.

#### Defect fixes

- Modified OEM SAS Enclosure NoDriver entry in LSIPSEUD.INF files.
  - Seen by OEM. Wrong PnP ID was given – OEM made a modification.
  - Root cause: Bad original PnP ID from OEM.
  - Fix: Changed to correct ID.
- In FwDiagPostImmediateBuffers function, call to PostFwDiagBuffer was checked for a FALSE return, but should be checked for a TRUE return. This would cause an invalid event log entry.
  - Seen by internal test.
  - Root cause: Check for 'if (!PostFwDiagBuffer...)'.
    - Fix: Changed to check for 'if (PostFwDiagBuffer...)'.
      - Fix: Changed to use U16 for loop counter.
- In FwDiagPostImmediateBuffers function, 'loop' counter is not big enough to account for maximum loop count.
  - Seen by internal test.
  - Root cause: Used U8 instead of U16 for loop counter.
  - Fix: Changed to use U16 for loop counter.
- Vista/Server 2008 drivers become uninstallable.
  - Seen by OEMs.
  - Root cause: .SYS files are not timestamped when signed.
    - Fix: Start timestamping .SYS files when Vista/Server 2008 drivers are signed.

- Modify OEM timeout value in registry.
  - Seen by OEM.
  - Root cause: Timeout value was not large enough to support some systems.
    - Fix: Changed INF files to increase timeout value from 65 to 160 seconds.

# Version 1.28.02

## Major Changes From Version 1.28.01:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Bluescreen while pulling drives in IR setup.
  - Seen by internal test.
  - Root cause: Upper-level driver (StorPort) can issue an escalated reset while driver is recovering from a hard reset. StorPort can re-use the same SRB in the escalated reset that was used in the previous reset even though the previous reset has not completed. The LSI driver queues the escalated reset, finishes the first reset, and releases the SRB back to the OS. When the escalated reset is executed from the reset queue, the system can bluescreen because the SRB for this reset has been released and may have garbage in it.
    - Fix: Check if the reset reply is the expected reset type. If not, the escalated reset is not issued and the reset queue is checked for another reset to issue. Also, check if driver is in recovery state from a hard reset and if it is, don't queue resets. Instead, return SRB\_STATUS\_ERROR to the OS.
- INF files are not consistent and some have incorrect OEM information.
  - Seen by OEM.
  - Fix: Change OEM sections of INF files to be correct.
- F/W diagnostic buffer IOCTLs do not return the proper error codes.
  - Seen by internal test.
  - Root cause: When issuing a F/W diagnostic buffer IOCTL with a properly formatted and sized buffer, the IOCTL will succeed, but the ReturnCode in the SRB\_IO\_CONTROL structure will be 2 instead of 0.

Also, in case of an error, a generic IOCTL error is returned instead of the SRB\_IO\_CONTROL ReturnCode.

- Fix: Return the proper error code and change the error code value of 2 to 3 so that applications can know that this is a real error. The old value of 2 will tell an application that the error can be ignored if using an older driver. Also, add a new ReturnCode so that if an error occurs the entire buffer is returned so that an application can view the IOCTL error code. If an application uses this new ReturnCode and the IOCTL ReturnCode upon completion is not successful, the entire buffer is returned.

# Version 1.28.01

## Major Changes From Version 1.27.03:

### General Changes

#### Functionality

- In addition to code 0xDEAD (diag reset), added a bluescreen with code 0xFACE when FW faults. The code to force the bluescreens was changed to work for released builds of the driver as well as checked builds. These bluescreens will occur only if the registry entry "StopOnDiagReset" is active.
- Updated to MPI headers 1.05.19.

#### Defect fixes

- None.

# Version 1.27.03

## Major Changes From Version 1.27.02:

### General Changes

#### Functionality

- Updated to MPI headers 1.05.18.

#### Defect fixes

- Modified OEM SAS Enclosure NoDriver entry in LSINODRV.INF files.
  - Seen by OEM. Wrong PnP ID was given – OEM made a modification.
  - Root cause: Bad original PnP ID from OEM.
    - Fix: Changed to correct ID.



# Version 1.27.02

## Major Changes From Version 1.27.01:

### General Changes

#### Functionality

- Added more OEM SAS Enclosure NoDriver entries to LSINODRV.INF files to suppress asking for drivers when these enclosures are present.
- When Persistency Table is full and Log Exception Events is enabled, an error is logged to the Windows System Event log.

#### Defect fixes

- Bluescreen when reply frame has HBA API flag set and original request did not.
  - Seen by OEM during Windows 7 testing.
  - Root cause: Bad reply frames are returned that have the HBA API flag set, even though the original request did not. When reply is processed, bluescreen occurs because there is no outstanding HBA API request.
    - Fix: Add checks in HbaApiWmiCompletion and CheckHbaApiScsilo to make sure the request buffer context also has the HBA API flag set.
- Bluescreen during handling of NULL LuExt to complete a context reply.
  - Seen by OEM during Windows 7 testing.
  - Root cause: Port driver issues I/O's to devices with no LuExt allocated, causing bluescreen.
    - Fix: Check for NULL LuExt before processing the reply.

# Version 1.27.01

## Major Changes From Version 1.26.05:

The Vista IA64 driver is no longer included in the Windows driver release package.

## General Changes

### Functionality

- Lowered the minimum accepted value for MaximumTargetQueueDepth from 8 to 1.

### Defect fixes

- Changed to return SRB\_STATUS\_INVALID\_REQUEST instead of SRB\_STATUS\_ERROR if an invalid control code was detected for an IOCTL call.
  - Seen by OEM.
  - This issue is seen only with third-party drivers. To reproduce, use one of these third-party drivers. A driver before this fix will take a long time (20 minutes) to boot.
  - Root cause: Wrong error code is returned for an invalid IOCTL control code.
    - Fix: Return proper error code for an invalid IOCTL control code.
- Changed LSImpiReset function to issue a Diag Reset if there is a stuck LUN or Target Reset, instead of a Message Unit Reset.
  - Testing and customer experience has shown that MUR does not always clear up the stuck reset condition.
  - Issue can only be seen if F/W has a bug where a LUN or target reset gets stuck. If Event 11's in the Windows System event log with the data at offset 0x10 of 0xAD010036 are seen, then LUN or target resets are getting stuck.
  - Root cause: F/W bug.

- Fix: Send Diag Reset instead of Message Unit Reset if a stuck LUN or Target Reset condition occurs.
- Changed ProcessTmResetReply function to issue a Diag Reset if there are still outstanding I/O's after issuing a Bus reset.
  - Seen by OEM running Vista WinPE with F/W version 1.23.04-IR.
  - Issue can only be seen if F/W has a bug where a bus reset does not return all outstanding I/O's on that bus.
    - Root cause: F/W bug.
  - Moved InterruptMgmtValues in SCSI Server 2008 IA64 .INF file to correct location.
    - Seen internally.
  - MPI 1.7 adds new functionality to the DeviceInfo field. Previously, the driver used some of these bits for internal flags. A mask was added to the driver to account for possible interference with the new MPI functionality.

# Version 1.26.05

## Major Changes From Version 1.26.04:

### General Changes

#### Functionality

- Added support for EEDP BYTCHK bit when set to 1 for Verify command (true data verification).

#### Defect fixes

- None.

# Version 1.26.04

## Major Changes From Version 1.26.03:

### General Changes

#### Functionality

- Added StopOnDiagReset registry entry to cause bluescreen on a Diag Reset for debugging purposes.

#### Defect fixes

- Fixed failed recovery handshake for PCIe adapters. Fixed problem where PCIe adapters share interrupts with other PCIe parts and this can cause a diag reset to fail. If another device interrupts but we share that interrupt, and we're outside of our recovery routine (waiting the 1/10 second to come back in), our ISR will get called and it will read the doorbell, then it will clear the IntStatus register. This "swallows" the data that is being returned via the doorbell handshake and the PortEnable fails.
  - Seen by OEM.
  - To reproduce: 1) Have multiple 949E chips in an IBM server box. 2) Upgrade firmware and do a chip reset. 3) LSIUTIL gets "stuck" without exit.
  - Root Cause: PCIe adapters share interrupts with other PCIe parts which can cause a diag reset to fail. If another device interrupts but on the shared interrupt, and driver is waiting the 1/10 second to come back for watchdog, ISR will get called and it will read the doorbell, then it will clear the IntStatus register. This "swallows" the data that is being returned via the doorbell handshake and the PortEnable fails.
  - Fix: Clear the Interrupt Status register only if IOC is not operational OR if the IOC is operational and doorbell interrupts are enabled.
- Added check for NULL SRB in CheckInqFlagReplies function. Previously, if NULL SRB was seen here this would cause a bluescreen.
  - Seen by OEM.

- Can only reproduce when firmware sends bad reply.
- Root cause: Firmware sends bad reply.
  - Fix: Check for NULL SRB, log error, and return from function without processing further.
- Moved InterruptMgmtValues\_nomsi in FC Server 2008 x86 .INF file to correct location.
  - Seen internally.

# Version 1.26.03

## Major Changes From Version 1.26.02:

### General Changes

#### Functionality

- Added Server 2008 production build support.

#### Defect fixes

- Changed OEM SAS SES PnP ID to remove F/W version.
  - Seen by OEM.
  - To reproduce: Update the OEM SAS SES device F/W to version 02.0. On a system install or reboot the New Hardware Found dialog box will be displayed. Select the option to allow the OS to search for a driver. No driver will be found and the SES device will have a yellow bang (!) in Device Manager.
  - Root Cause: The OEM SAS SES PnP ID in LSIPSEUD.INF contained the F/W revision.
  - Fix: Removed the last 4 characters (F/W revision) from the PnP ID.

# Version 1.26.02

## Major Changes From Version 1.26.01:

### General Changes

#### Functionality

- Added OEM SAS Enclosure NoDriver entries to LSINODRV.INF files to suppress asking for drivers when these enclosures are present.

#### Defect fixes

- None.



# Version 1.26.01

## Major Changes From Version 1.25.11:

### General Changes

#### Functionality

- Updated to MPI headers 1.05.17.
- Added support for detecting dual port SAS drives or detecting SATA drives behind a port selector.
- Added a reset queue so that multiple resets occurring at the same time for different targets or different target/LUNs can be queued up and still executed in the order received.
- Changed handling of F/W diagnostic buffers such that when a diag reset or message unit reset is done a posted buffer is changed to be released and is available for the diag app to retrieve it.
- FC HBA API was changed to accommodate Vista during WHQL testing.

#### Defect fixes

- Provide workaround for DMA transfers using 1078 HBA on systems with more than 36 GB of memory.
  - Seen in OEM lab. Reproduce by writing data to at least 10 files at one time.
    - Symptom: System hangs during DMA transfers.
    - Root cause: DMA transfers while using 1078 HBA on systems with more than 36 GB of memory.
    - Fix: Walk through SGL to find problematic addresses. If bad address is found, use POM2 addressing instead of POM3 addressing. Bad addresses are in the ranges:  
0x00000009\_00000000 through 0x00000009\_1000FFFF  
0x00000009\_18000000 through 0x00000009\_18000FFF  
0x00000009\_1C000000 through 0x00000009\_1C000FFF

This fix checks for the range:

0x00000009\_00000000 through 0x00000009\_FFFFFFFF

- Fixed bus/target persistency mapping problem when using generic IOCTL calls or Task Management calls issued through the MPI IOCTL interface. This affects Fibre Channel controllers only.
  - Seen in OEM lab.
  - Symptom: Wrong data will be returned when issuing generic IOCTL calls when host driver is in LUN persistency mapping mode.
  - Root cause: Host driver is not remapping bus/target addresses during these calls.
  - Fix: During generic IOCTL calls or task management commands issued through IOCTL interface, remap bus/target addresses if using LUN persistency mapping.

# Version 1.25.11

## Major Changes From Version 1.25.10:

### General Changes

#### Functionality

#### Defect fixes

- Increase timeout after MPI\_DIAG\_RESET because downloading Phase 9 firmware (or later) takes longer than old timeout of 30 seconds, which causes system hang. Increased timeout to 300 seconds.
  - Detected by OEM
  - Symptom: After downloading Phase 9 firmware (or later) on top of the bridge firmware the system will hang after SASFlash issues the reset.
  - Root cause: The timeout during the reset in the driver was not long enough to account for the longer reset time from phase 9 onward.
  - Fix: Increase timeout from 30 to 300 seconds.

# Version 1.25.10

## Major Changes From Version 1.25.09:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed looping for getting the NonCachedExtension
  - Detected via code review
  - Symptom: If registry settings are used to enable usage of F/W diagnostic buffers, and the requested buffer size allocations are too large for the OS to support, the Non-Cached Extension allocation will fail. The miniport should loop back and attempt the Non-Cached Extension allocation again to allow the system to boot, but due to a logic error the system will hang.
  - Root cause: A loop flag was not cleared properly on the 2<sup>nd</sup> allocation attempt.
  - Fix: Clear the loop flag at the start of the loop.
- Fixed FC HBA API for updated DTM tests
  - Seen in OEM labs, reproduced in LSI lab
  - To reproduce: Run the DTM WHQL tests for FC adapters. The FC HBA API test will fail.
  - Root cause: The FC HBA API test has been updated in the DTM WHQL tests. Additional test cases were added which expected the miniport to check SCSI command return data (the data buffer) for additional data lengths. Also, the SendCTPassThru test failed because a test script error treats the MaximumSize field in the CT\_IU preamble as little endian instead of big endian.
  - Fix: Add code to ScsiReportLuns and ScsiInquiry tests to check return data buffers and detect additional data available. Added a workaround

for bad MaximumSize field to set the field to the maximum miniport buffer size.

# Version 1.25.09

## Major Changes From Version 1.25.08:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fix for issuing Task Management resets via IOCTLS.
  - Seen by LSI test lab.
  - To reproduce: With all drives online and no IO running, issue a Target Reset to any one of the drives connected to the HAB using LSIUtil. LSIUtil reports “Failure to Issue Command”. After a few seconds, the HAB resets, Windows Blue-Screens with Stop 0x0D1 and reports an error with Storport.sys. This issue does not occur in Limpopo.
  - Root cause: Due to a change made in 1.25.06 to avoid multiple calls to SynchronizeAccess.
  - Fix: Back out change in 1.25.06 for multiple SynchronizeAccess calls. Instead don't call SynchronizeAccess if a dual-channel device needs to restart a channel.
- Fix misaligned data problem on IA64 machines.
  - Seen in Microsoft labs when using MSI/MSI-X only.
  - To reproduce: On IA64 system, when CheckEnableMSIX is called causes a machine exception bugcheck. Problem does not happen on x86 or x64 systems because these systems can handle these misaligned accesses..
  - Root cause: When using MSI/MSI-X on IA64 system, data is misaligned.
  - Fix: Move PciConfigSpace array so that it is dword/quadword aligned.

# Version 1.25.08

## Major Changes From Version 1.25.07:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed reported status of an inactive volume to be 'Offline' (was reporting OK).
  - Seen by LSI test lab.
  - To reproduce: Roam an IR volume to a different HBA (causing the volume to be Inactive). Run the GetRaidConfig IOCTL. Note that the Inactive volume is show with an OK status.
  - Root cause: The check for an inactive volume was done after the check for an OK status and was not being executed.
  - Fix: Moved check for Inactive volume to first check of volume status.
- Fixed MSI-X vector reset issue on resume where F/W is not Ready and KickStart is called (save/restore MSI-X vector info).
  - Seen in Microsoft labs. (Only on Server 2008)
  - To reproduce: Run Server 2008 interim build on SAS HBA and perform Standby test. On resume, I/O's to HBA will all time out due to MSI interrupts not being propogated.
  - Root cause: On resume, miniport checks the F/W state and if not in the Ready state a hard reset of the adapter is performed. On some systems, even though the SAS HBA has F/W in flash, the F/W does not get to the Ready state by the time this check is done. The hard reset of the HBA at this time causes the MSI-X vector table to be reset (interrupts disabled).
  - Fix: In the KickStart routine (adapter hard reset), if MSI-X is enabled, save off the MSI-X vector data, perform the hard reset, then restore the MSI-X vector data.

# Version 1.25.07

## Major Changes From Version 1.25.06:

### General Changes

#### Functionality

- Remove requirement for Port I/O access (mainly 1030 FWDLB), but don't enable via registry entry. (see Special Notes above)
- Enable MSI/MSI-X support controlled by #define MSI\_ENABLED.  
(Disabled for all OS versions up to and including Vista. Will be enabled for Server 2008.)

#### Defect fixes

- Backed out workaround to fail LUN or target resets if no I/O's were terminated by the reset (caused clustering problems).
  - Original workaround was due to issues seen during Vista beta test cycles at Microsoft. Recent discussions with Microsoft indicate this workaround should not be necessary.
  - Clustering solutions will issue a LUN reset to break reservations. Failing the LUN reset because no I/O's were terminated caused escalation to target and bus resets, which are disruptive to other storage devices.
  - Result of this change is that a LUN reset issued to break reservations will return with successful status, even if no I/O's are terminated.
- Fixed ISR to complete ALL entries on ReplyFIFO (if MSI enabled)
  - Issues seen in Microsoft test labs when running with MSI/MSI-X enabled on Server 2008 beta releases (commands not completing properly).
  - Root cause: A logic error in the driver ISR could result in not all pending interrupts being handled before leaving the ISR (required for MSI/MSI-X).



- Fix is to process ALL pending interrupts before leaving the ISR.  
(Note: This does not affect any current LSI production drivers. It only affects Server 2008 drivers running with MSI/MSI-X enabled.)
- Fixed FC HBA API SendCTPassthru DMA problem (handle non-contiguous data buffers).
  - Seen at Microsoft Server 2008 test labs, reproduced by LSI.
  - To reproduce, issue a Fibre Channel Common Transport message via the FC HBA API interface with a response buffer size that crosses a non-contiguous page boundary. The data will be DMA'ed into contiguous memory starting with the first page.
  - Root cause: The driver built only one SG entry each for the request and response buffers (assumed contiguous memory in IOCTL buffer).
  - Fix: The request buffer is copied to a local contiguous buffer. This is required since the MPI F/W interface for issuing a CT command allows only one SG entry for the request buffer. This buffer size is limited to 2048 bytes. Build a full SG list for the response buffer.
- Updated LSINODRV.INF files for Vista with a DriverVer date later than 1/1/07 (WHQL Signability test failures)
  - WHQL Signability test (part of the DTM test suite) will fail if a DriverVer date is earlier than 8/21/2006.
  - DriverVer dates for the LSINODRV.INF files for Vista were updated to dates later than 1/1/2007.

# Version 1.25.06

## Major Changes From Version 1.25.05:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Added check for non-zero DeviceExtension in FindAdapter, zero out if necessary (recent change in StorPort can result in DeviceExtension being non-zero).
  - Seen in Microsoft lab with Longhorn beta release testing.
  - Due to a change in the StorPort port driver, under certain power management scenarios the port driver can call the LSI FindAdapter routine with a reused DeviceExtension. According to DDK documentation the DeviceExtension is always zero'ed out on a call to FindAdapter.
  - After discussions with Microsoft this new behavior will remain in the StorPort driver, therefore a check was added to FindAdapter and the miniport driver will zero it out if necessary.
- Fixed possible hang due to multiple SynchronizeAccess calls when issuing IOCTLs. (Applies only to multi-function adapters, such as Fibre Channel or U320. Does not apply to SAS adapters.)
  - Seen in LSI test lab on FC adapters.
  - During diagnostic reset testing calls to IOCTL routines after a Diag Reset IOCTL could result in a system hang condition.
  - Root cause: A reset on the opposite channel could cause multiple calls to the StorPortSynchronizeAccess routine, which results in a deadlock condition.
  - Fix: Modify IOCTL queuing routine to return from the SynchronizeAccess call before doing an immediate issue of an IOCTL.

# Version 1.25.05

## Major Changes From Version 1.25.04:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Added BlockISR flag to prevent H/W accesses by our ISR whenever we are shutdown (workaround StorPort bug).
  - Seen by OEM test labs, LSI lab, Microsoft lab.
  - To reproduce: Install StorPort driver version 5.2.3790.4021 or later, LSI HBA with a shared IRQ line with one or more other devices. On system shutdown or driver upgrade the system will get a 0x9c Machine Check Exception bluescreen.
  - Root cause: A change in the above version of the StorPort driver allows the LSI driver interrupt service routine (ISR) to be called even after the LSI adapter has been shutdown and unmapped from PCI I/O and mem space. The ISR attempts to access H/W registers as a machine check exception results.
  - Workaround added: A flag was created that is clear on driver init, set on adapter shutdown, and cleared again on adapter resume. This flag is checked first thing in the ISR and if set the ISR just returns FALSE, indicating this is not our interrupt.

NOTE: The Server 2003 Service Pack 2 version of StorPort also has this bug.
- Modified handling of SAS device removal events to issue only one StorPortPause/Resume pair and only one BusChangeDetected notification (workaround StorPort bug).
  - Seen in LSI labs
  - To reproduce: Install Server 2003 Service Pack 2. Attach a large SAS topology (60+ drives) through a single wide-port link to the SAS

initiator. While system is running, pull the wide-port cable from the initiator. The failure is that none of the drives are removed from the OS, and if Disk Management is run it will hang trying to rescan for devices.

– Root cause: A change in the Service Pack 2 version of the StorPort driver causes the large number of StorPortPause/StorPortResume calls and the large number of BusChangeDetected notifications to block the issuing of a bus rescan. This prevents the OS from detecting that the disks have been removed and also causes Disk Management to hang.

Fix: The SAS device removal routine was modified to issue only one StorPortPause at the beginning of processing the multiple removals, then only one StorPortResume when all removal processing was completed, and only one BusChangeDetected notification at the end of processing.

# Version 1.25.04

## Major Changes From Version 1.25.03:

### General Changes

#### Functionality

- Added SAS Broadcast Async Event support.

#### Defect fixes

- None.

# Version 1.25.03

## Major Changes From Version 1.25.02:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Inquiry commands issued to a particular OEM SAS tape drive that returned Check Condition status would have no Sense Data returned.
  - Seen in OEM customer lab, reproduced by LSI.
  - Issue a mal-formed Inquiry command (bad byte in CDB) to the OEM SAS tape drive and note that Check Condition status is returned by no Sense Data is returned.
  - Root cause: Inquiry commands to this OEM SAS tape drive go through special processing to enable SAS TLR. This routine was not detecting a bad SCSI status on an Inquiry command so the Sense Data was not transferred to the OS I/O structure.
  - Fix: Added a check for a bad SCSI status. If so, forward I/O processing to the SCSI IO error reply handling routine.
- Internal Task Management Target Reset changes.
  - Seen in drive manufacturer test lab.
  - During drive power cycle and hot plug testing, the F/W will issue a SAS Device Status Change Event with a Device Not Responding reason code. When the driver receives this event, it must issue a TM Target Reset to help clean up any outstanding I/Os. The drive manufacturer observed that in some cases a 2 second timeout on handshaking the TM message was too short (F/W could take up to 4 seconds to respond to the handshake.) Also, in some cases the TM reset reply was never received, resulting in OS I/O's being blocked forever.
  - Fixes: 1) Increased reset message handshake timeout from 2 to 5 seconds. 2) Added setting of the ResetScsiTimer on these internal TM

Target Resets. If a TM reply is not received within 15 seconds, the driver will hard reset the controller.

# Version 1.25.02

## Major Changes From Version 1.24.04:

Version numbers from 1.24.05 through 1.25.01 were skipped to account for the driver packaging restructuring.

## General Changes

### Functionality

- Send BusChangeDetected notifications if the chip is reset due to a F/W fault (IR device changes may go unscanned).
  - Seen by OEM customer, reproduced in LSI labs.
  - With certain SAS F/W, an IR volume creation will result in a F/W fault before the Volume Create event is issued to the host driver. The OS not seeing the newly created volume since a bus rescan was not performed.
  - Workaround: At any time the host driver detects a F/W fault and will hard reset the chip, issue a BusChangeDetected for all busses of the adapter. This will cause the OS to rescan each bus and any drive changes that occurred before the F/W fault will be detected.
- Added check for invalid NVRAM (bad checksum).
  - Seen by OEM customer, reproduced in LSI labs.
  - OEM System with invalid NVRAM (checksum errors) would not initialize U320 adapter properly (driver initialization would fail).
  - Fix: Added check for invalid NVRAM error return and allow initialization to continue. This results in a NULL HBA serial number being reported by custom OEM IOCTLs.
- Added support for surprise removal of the adapter.
  - New requirement for Windows Longhorn testing
  - All PCI register reads are checked for return value of 0xFFFFFFFF (adapter not responding). A BusChangeDetected notification is sent and a status of NO\_HBA is returned on all subsequent commands.

### Defect fixes



- Add IR PhysDiskCreated/Deleted reason codes to BusChangeDetected notification test (detect change due to create/delete hot spare).
  - Seen by OEM customer, reproduced in LSI labs.
  - If a hot spare drive was created or deleted, the associated OS availability for the physical drive was not detected.
  - Fix: Added the PhysDiskCreated/Deleted reason codes to the check for when to issue a BusChangeDetected.
- Modify proprietary MSG\_DIAG\_RESET IOCTL to avoid a system stall during adapter reset.
  - Seen in LSI labs.
  - During a F/W download using Sasflash, the mouse and keyboard would be unresponsive for 20 to 30 seconds.
  - Driver processing of the MSG\_DIAG\_RESET IOCTL (performs a hard adapter reset) was being handled all in-line, blocking lower level interrupts.
  - Fix: Modified DelayuedDiagResetIoctl routine to utilize the timer recovery routines (replacing SynchronizeDiagResetIoctl). This performs the reset in small increments, without delay loops.
- Increase default number of reply frames to MaxDevices + 1 for SAS. (Fix starving of reply frames on removing big topologies.)
  - Seen by OEM customer, reproduced in LSI labs.
  - If more than 64 target devices are removed from the SAS topology in one action (such as removing a wide SAS connection from cascaded expanders/targets) the devices are never removed from the OS.
  - Root cause is that the F/W expects to be able to issue all device removal events before processing any EventAck commands. The driver would allocated only 64 reply frames by default.
  - Fix: Increase the default number of reply frames to MaxDevices plus 1 (but limited by the F/W ReplyQueue Depth).

# Version 1.24.04

## Major Changes From Version 1.24.03:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed handling of SenseInfoBuffer addresses that are above the lowest 4GB for SCSI I/O's issued through SendIOCTLMessage and custom OEM IOCTLs.
  - Seen by end customers, OEM customer, reproduced in LSI lab.
  - To reproduce the following environment is needed:
    - System is running an LSI\_SCSI, LSI\_SAS, or LSI\_FC driver with a version number up to and including 1.24.03.
    - System is configured and is using more than 4GB of memory.
    - An application, agent, or service is calling the LSI proprietary MPI generic IOCTL interface.
    - The MPI messages being issued through the MPI generic IOCTL interface include SCSI IO or SCSI RAID Passthru IO.
    - The SCSI IO or SCSI RAID Passthru IO command receives a check condition and sense data from the target device
  - If all of the above conditions are met it is possible that the LSI driver will cause memory corruption. For this to occur, the IOCTL Srb DataBuffer must reside in physical memory that is above the lowest 4GB boundary.
  - Root cause: A logic error in the driver ignores the upper 32-bits of the physical address of the DataBuffer and will DMA the sense data to the truncated lower 32-bit address.

- Fix: A local buffer (always in the lowest 4GB of memory) is used to DMA sense data, then that buffer is copied to the IOCTL SrbDataBuffer.

# Version 1.24.03

## Major Changes From Version 1.24.02:

### General Changes

#### Functionality

- Due to a F/W workaround, I/O's can be completed with an IO\_CANCELLED\_DUE\_TO\_R\_ERR IOCLogInfo code for retry by the OS. This IOCLogInfo code was added to the list to be ignored, so an Event 11 would not be logged in the Windows Event Log.
- Changed RequestsToComplete parameter in StorPortBusy call to 10 (best performance). Performance testing showed that a fixed value of 10 I/O's results in the highest performance, rather than waiting for 10% of the outstanding I/O's to complete. (StorPortBusy is called when all F/W credits are exhausted by outstanding I/O's.)

#### Defect fixes

- Added pausing of adapter when F/W fault detected in WatchdogTimer routine (also added code to ensure adapter is unpaused after a hard reset recovery).
  - Seen by LSI test lab.
  - In some situations, when a F/W fault occurred the system would hang forever.
  - Root cause was that the adapter was not being paused when the fault was detected. This allowed additional I/O's to be received while a hard reset was being done. These would time out due to the hard reset and an OS reset would be issued, which would pause the adapter. The adapter was not getting unpaused when the hard reset completed.
  - Fix: Pause the adapter as soon as the F/W fault is detected. Also, ensure that no additional pauses are issued, and that the adapter is unpaused after the hard reset is completed.

# Version 1.24.02

## Major Changes From Version 1.24.01:

### General Changes

#### Functionality

- Updated INF and txtsetup.oem files with custom OEM 1068E branding entries.
- Added SAS x28 and x36 Expander NoDriver entries (to suppress asking for drivers when LSI SAS expanders are present).

#### Defect fixes

- Fixed problem with SendIOCTLMessage chain element being placed beyond end of message buffer (Sasflash error on x64 systems.)
  - Seen by customer, reproduced by LSI test lab
  - Running “sasflash -b mptsas.rom” on a Windows x64 system would not complete an update of the SAS HBA BIOS.
  - Root cause was the building of the MPI SG list for the F/W Upload command was placing the chain element beyond the end of the request buffer.
  - Fix: Changed the calculation for maximum number of SG elements in the request message to account for a possible chain element at the end of the buffer.
- Fixed check for other errors along with a data underrun on an Inquiry command in CheckInqFlagReplies.
  - Seen by LSI developer.
  - If a device returns a check condition on an Inquiry to a non-existent LUN (instead of returning Inquiry data), the LUN would show up in Device Manager as just “SCSI Device”.
  - Root cause was treating a data underrun status with another error as a success status.
  - Fix: If another SCSI error status exists along with the data underrun, process the error status.

- Fixed adding of EEDP protection bits in BuildScsilo32. MS port drivers still put LUN number in CDB byte 1, needs to be masked. (applies to Fibre Channel devices only)
  - Seen by LSI developer.
  - EEDP SCSI IO 32 message frames would not have the proper protection bits set for LUNs other than LUN 0.
  - Root cause is that Microsoft port drivers still insert the LUN value into byte 1 of the CDB. That is now where the protection bits reside.
  - Fix: Mask off the absolute LUN field in byte 1 before OR'ing in the EEDP protection bits.

# Version 1.24.01

## Major Changes From Version 1.21.30:

Versions 1.22.xx were reserved. Versions 1.23.xx were used for Vista/Longhorn bundled drivers.

## General Changes

### Functionality

- Added support for Vista/Longhorn (many changes to HBA shutdown/recovery routines). (Driver installation packages have not been checked in yet.)
- Added support for multiple hot spare drives. (Starting with SAS F/W 20.04 two hot spares are supported.)
- Modify test for hot spare (GetHotSpare) to check vendor field to determine drive type for a missing hot spare drive. (Allows missing hot spare drive to be displayed in GetRaidConfig even after a reboot of the system.)
- Add logging of Windows event for Fibre Channel if link speed changes (can be disabled by registry entry).

### Defect fixes

- Fixed problem with GetRaidConfig options Device ID and Additional Data when volume is resyncing.
  - Discovered by LSI developer.
  - Symptom was running CSMTTest, option 6, suboptions 2 or 3, output would display many lines of “RAID Config Number x”, where x is incrementing.
  - Root cause was for suboptions 2 and 3 (when resyncing), GetRaidConfig was not filling in the bDriveCount element before returning the IOCTL.
  - Fix: Update bDriveCount before returning IOCTL.

- Added checks on DomainValidation code to protect from rogue replies (parallel SCSI only).
  - Seen in Microsoft labs.
  - Very intermittently, during ACPI Stress tests a reply would be seen that would have a Domain Validation flag bit set in the message context, even though Domain Validation was not in progress. Result would be a system bluescreen
  - Fix: Additional checks added to ensure that Domain Validation is actually in process before trying to dereference the DvSrb.
- For LsiWmiSendCTPassThru, adjusted minimum output buffer size to be 1052 bytes (12 bytes for parameters plus 16 bytes of preamble plus 1024 bytes output). Also, added checks for CT preamble MaximumSize field.
  - Seen in Microsoft labs, reproduced by LSI
  - Using the FC HBA API, send a CT passthrough with a very small response buffer. With older LSI FC929 F/W the CT passthrough will never complete.
  - Fix: Increase minimum application buffer size for CT passthrough to 1052 bytes, to allow for a 1024 byte response buffer length. This also involved adding checks of the CT preamble to ensure the CT message is indicating the correct response buffer size.



# Version 1.21.30

## Major Changes From Version 1.21.29:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed assignment of IOCsTotalDevices for SAS. Due to F/W mapping changes the PortSCSIID field must be used instead of MaxDevices
  - Discovered by developer code review.
  - The maximum number of target IDs that must be scanned by the OS was incorrect. It was possible for target IDs to be mapped that would not be seen by the OS.
  - Root cause: Starting with SAS IT F/W 10.14 the mapping of devices was changed to reserve sections of target IDs, resulting in the PortSCSIID being a higher value than 127 (MaxDevices).
  - Fix: Change from using MaxDevices field to using PortSCSIID as the largest mapped value that can be returned by the F/W (for SAS only).
- Fixed sizing of SAS address table due to F/W mapping changes.
  - Discovered by developer code review.
  - SAS address table was hard coded to a size of 130 entries. With the changing of the F/W mapping (see above issue) target ID can go well above 130.
  - Fix: Size of the SAS address table is now calculated using the IOCsTotalDevices value.

# Version 1.21.29

## Major Changes From Version 1.21.28:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed buffer length check in GetRaidConfig custom OEM IOCTL.
  - Seen by customer. Reproduced by LSI.
  - To reproduce, configure IS volume with 8 drives. Run dumpsas option /rc. System will return a -1 error code instead of displaying the RAID configuration information for the 8 drives.
  - Root cause: GetRaidConfig IOCTL was including the ioctlHeader in the minimum buffer length calculation, when it should not.
  - Fix: Removed ioctlHeader from the length calculation.

# Version 1.21.28

## Major Changes From Version 1.21.27:

### General Changes

#### Functionality

- Changed driver installation amd64 folder names to x64. LSI was requested by OEM customer to change the folder names to the generic x64 designation, as this denotes both AMD64 and EM64T systems.

#### Defect fixes

- Fixed drive display error in GetRaidConfig custom OEM IOCTL.
  - Seen by customer. Reproduced by LSI.
  - To reproduce, configure IM volume in Optimal state with a hot spare drive. Run CSMI IOCTL test GetRaidConfig, option 1) Drives. System will display all 3 drives correctly the first time, but will display all zeros for the first drive on subsequent runs.
  - Root cause: GetRaidConfig IOCTL was using an index before it was initialized.
  - Fix: Moved initialization of index to location before it is used.
- Fixed display of RAID volume capacity for volumes of greater than 2 TB capacity.
  - Seen by customer. Reproduced by LSI.
  - To reproduce, configure RAID volume with > 2TB capacity. Issue GetRaidConfig IOCTL. Volume capacity will be incorrect (much smaller than actual).
  - Root cause: RAID volume capacity calculation was not using the MaxLBAhigh field of RAID Volume Page 0
  - Fix: Modified calculation to use MaxLBA and MaxLBAhigh fields.
- Fixed GetLocation IOCTL when a RAID volume physical disk is missing.
  - Seen by customer. Reproduced by LSI.
  - To reproduce, configure IM volume. Issue GetLocation IOCTL. Both physical disks will have location data. Remove one of the RAID physical

disks. Issue GetLocation IOCTL. IOCTL will be returned with a Failed status.

- Root cause: A config page access for the missing physical disk was returning an error, causing the IOCTL to be returned with Failed status.
- Fix: Check physical disk status for a missing status. If missing, skip getting location data for this disk and go on to next disk.

# Version 1.21.27

## Major Changes From Version 1.21.26:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed errors in GetRaidConfig custom OEM IOCTL.
  - Seen by customer. Reproduced by LSI.
  - To reproduce, configure IM volume in Optimal state. Remove one drive to put volume in Degraded state. Add a new drive to replace removed drive, volume will begin resyncing. Run CSMI IOCTL test GetRaidConfig, option 1) Drives. System will either bluescreen or data for many drives will be displayed
  - Root cause: GetRaidConfig IOCTL was using an uninitialized pointer. Also, callback for GetHotSpare routine was not getting called correctly.
  - Fix: Moved HpSasGetHotSpare callback check to top of list (so it is called before other callbacks). Fixed uninitialized RAID Vol Page 0 pointer in GetRaidConfig.

# Version 1.21.26

## Major Changes From Version 1.21.25:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Modify check for PortSCSIID > SYM\_MAX\_TARGETS. Log error only for parallel SCSI. For FC or SAS if check fails set initiator ID to 255.
  - Reported by LSI test lab. With SAS IT F/W that supports > 126 target devices, an Event 11 with offset 0x10 data of 0xAD010019 will be logged on every boot.
  - Check for PortSCSIID needs to be made only for parallel SCSI, where the Initiator ID is used in the bus protocol.
  - Fix is to log the event only for parallel SCSI. For SAS and FC, if the PortSCSIID is > 255, set the OS Initiator ID to 255.
- Fixed GetRaidConfig, GetLocation, and GetRaidElement IOCTLs to check for proper hot spare compatibility.
  - Reported by OEM customer. Information for hot spare disks for these IOCTLs was not being reported.
  - Fix is to use additional checks of hot spare data in MPI config pages to obtain hot spare info and verify compatibility with the IR volume.
- Fixed GetLocation IOCTL to return location info for non-IR SATA disks.
  - Reported by OEM customer. GetLocation data was not returned for a non-IR SATA drive.
  - Fix is to add the DeviceInfo mask for SATA\_DEVICE to all checks for target devices in the driver internal SAS mapping table.

# Version 1.21.25

## Major Changes From Version 1.21.24:

### General Changes

#### Functionality

- Added preliminary SM\_HBA (SAS HBA API) support. Five elements were implemented: AdapterInformationQuery, GetPortType, GetAdapterPortAttributes, GetPortAttributesByWWN, and SM\_RefreshInformation.

#### Defect fixes

- Subtract 1 from NumberOfDiscoveredPorts in FillPortAttributes (FC HBA API).
  - Seen by OEM customer. Run FC HBA API application and observe that the NumberOfDiscoveredPorts for the LSI FC HBA is 1 more than what is expected.
  - Root cause is that LSI FC F/W logs into itself (the initiator) and the initiator is counted as a discovered port. FC HBA API applications don't expect that.
  - Fix is to subtract one from the number of discovered ports reported in FC Port Page 0.
- Added code to prevent simultaneous shutdown, which causes F/W download boot to fail on flashless systems.
  - Seen by OEM customer, reproduced in LSI lab. Symptom is that hibernation will fail.
  - Root cause: On some systems, the StorPort driver will shutdown both channels of a multi-function adapter at the same time. With 1030 flashless systems, this causes both channels to do a F/W download boot at the same time. This causes a collision between the two channels and the F/W download boot fails resulting in no F/W running for the hibernation copy of the driver.
  - Fix is to use the ResetHistory bit in the diagnostic register to detect

when the “other” side has initiated a reset (and F/W download boot) and just wait for that reset to complete (wait for F/W Ready state).

- Added setting of device queue depth for every LUN found.
  - Observed in LSI lab. Symptom is that no matter what queue depth of I/O's are sent from an application, a maximum of queue depth of 20 is seen on target devices.
  - Root cause is that StorPort sets a default queue depth for every device at 20. A higher queue depth can result in better I/O performance for certain workloads.
  - Fix is to use the StorPortSetMaximumQueueDepth call for every LUN that is detected (via Inquiry commands). Driver default value is 64 (63 for parallel SCSI), tunable via a registry entry.
- When IOC credits are exhausted, replace sending back of Busy with StorPortBusy call to pause I/O's until 10% of credits are returned.
  - Observed in LSI lab. To reproduce, attach a large (> 25) number of drives to an HBA and issue I/Os at a queue depth > 20. Symptom will be one or more CPU's will go to 100% utilization and the system will become unresponsive (mouse and keyboard).
  - Root cause is that the IOC credits are all used by outstanding I/Os and the driver was returning SRB\_BUSY status. With StorPort, it turns around a Busy I/O immediately and reissues it to the miniport. This results in a very tight loop, consuming CPU cycles.
  - Fix is to utilize the StorPortBusy call to temporarily halt issuing of I/Os. When IOC credits are exhausted the driver calls this routine with a parameter of how many I/O's to allow to complete before restarting the issuing of I/Os. The driver uses a completion count of 10% of total credits before resuming.



# Version 1.21.24

## Major Changes From Version 1.21.23:

### General Changes

#### Functionality

- INF File Update Only
- Add custom OEM INF entries for branded 1068E entries
- Update/add INF files for OEM SAS SES No Driver support.

#### Defect fixes

- None.

# Version 1.21.23

## Major Changes From Version 1.21.22:

### General Changes

#### Functionality

- Added support for CSMI IOCTLS 0.90 (Phases 1 & 2).

#### Defect fixes

- Fixed FwDiagPostImmediateBuffers to read the reply frame for each post.
  - Seen by customer with their customized driver. If initial F/W Diag Buffers are enabled via the registry, subsequent messages would receive confusing reply frames.
  - Root cause was that the driver was posting initial diag buffers, but not reading the reply frames from those posts.
  - Fix was to add code in the initial buffer post routine to read the reply frames from the post messages.
- Added workaround for 106xE D3 to D0 transition issue (doorbell handshake would fail).
  - Seen everywhere when doing power management or driver upgrades with 106xE devices. Adapter would not be functional after a power transition or a driver upgrade.
  - Root cause is a H/W errata which leaves the doorbell reading Ready after a PCI D3 to D0 power state transition. This allows the host driver to perform doorbell accesses before the hardware and firmware are actually Ready.
  - Workaround is to perform a diagnostic reset on the device during driver initialization (start-of-day) and when resuming from any power transition state.
- Fixed hibernation failure introduced in 1.21.22.
  - Found in LSI test lab. Hibernation would fail or hang.
  - Root cause is the default setting of host persistency active would

cause the driver to try to allocate too much non-paged pool (NonCached Extension) and driver init would fail.

- Fix is to not set host persistency on in crash dump mode.

# Version 1.21.22

## Major Changes From Version 1.21.21:

### General Changes

#### Functionality

- Changed the default for Fibre Channel LUN Persistency from OFF to ON.

#### Defect fixes

- Fix to avoid double completion of lun/target resets.
  - Seen in LSI test lab.
  - During FC cable break testing, system can bluescreen due to a double completion of a LUN or target reset issued by StorPort. Investigation showed that StorPort is placing the reset on the active queue before it is issued to the miniport. This can result in another thread (our ISR thread) completing the reset via a CompleteRequest call during cleanup after a diagnostic reset of the controller. This is a synchronization problem within StorPort and will be escalated to Microsoft.
  - As a workaround, always check the SrbExtension for NULL before acting on a LUN or target reset (either issuing or completing). The SrbExtension is set to NULL when StorPort completes the Srb.

# Version 1.21.21

## Major Changes From Version 1.21.20:

### General Changes

#### Functionality

- None

#### Defect fixes

- Backed out CSMI 0.89 IOCTL changes due to improper header file structure definitions. Restoring CSMI 0.83 functionality.
  - Seen in LSI labs and reported by OEM (owner of CSMI header file).
  - Certain CSMI IOCTLs would fail due to application buffer length being too small.
  - New structure definitions caused compiler-generated padding, resulting in larger buffer sizes than implicit structure definition
  - Fix is to revert back to previous header file until fixed definitions are available.

# Version 1.21.20

## Major Changes From Version 1.21.19:

### General Changes

#### Functionality

- Changed GetRaidConfig IOCTL to get metadata size from IOC Page 6, if available.
- Changed GetConnectorInfo IOCTL to retrieve information from Manufacturing Page 7.
- Added fields to GetRaidInfo and GetRaidConfig IOCTLs. Added GetRaidFeatures and SetRaidControl IOCTLs.

#### Defect fixes

- Fixed erroneous access to RAID Physdisk Page 0 with IT firmware
  - Found in LSI test lab.
  - Request for RAID Physdisk Page 0 header without checking if the page was supported by the firmware resulted in an event in the Windows system event log.
  - Changed code to not request the RAID Physdisk Page 0 header if the current firmware is IT.

# Version 1.21.19

## Major Changes From Version 1.21.18:

### General Changes

#### Functionality

- Removed support for 1066 and 1066E.

#### Defect fixes

- Added checks to not request IOC Pages 3–6, Raid Volume Pages 0–1, and Raid Phys Disk Pages 0–1 for IT firmware.
  - Seen by LSI developer.
  - Due to limited space, firmware has removed support for the above pages in IT firmware. Requesting these pages in non-IR firmware results in incorrect results for various IOCTLs.
  - Fix: Check IOC Page 2 capabilities flag for Raid support in the firmware. If Raid is not supported, do not request the unsupported pages.
- Modified TM Reset replies to return an error status if no I/O's are terminated on a LUN reset.
  - Seen in LSI test lab.
  - For FC, when a TM Lun reset completed but no I/O's were terminated, the Srb would be returned with Success and result in a bluescreen.
  - Fix: Return the Srb with an error status when no I/O's are terminated for a TM Lun reset.

# Version 1.21.18

## Major Changes From Version 1.21.17:

### General Changes

#### Functionality

- None

#### Defect fixes

- Added checks to not request a rescan on certain SAS Device Status Change events.
  - Seen by LSI developer.
  - When certain SAS Device Change Events were received (such as ReasonCode INTERNAL\_DEVICE\_RESET, a rescan was being requested, but should not be. This results in unnecessary bus rescans.
  - Fix: Add checks to ensure that only ReasonCodes that require a bus rescan request one.



# Version 1.21.17

## Major Changes From Version 1.21.16:

### General Changes

#### Functionality

- Added support for SAS nexus loss delay functionality.

#### Defect fixes

- Added check for NULL LuExt in CheckInqFlagReplies.
  - Seen in LSI test lab. Issue of Inquiry command which isn't a port driver enumeration request to a non-existent device would result in a bluescreen.
  - ScsiPortGetLuExtension was returning NULL since device didn't exist.
  - Fix is to check for NULL LuExt and exit routine to let ISR complete command.

# Version 1.21.16

## Major Changes From Version 1.21.15:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Fixed how the GetRaidConfig ioctl retrieves a disk serial number.
  - Seen by customer. Significant characters are truncated from the drive serial number while trailing blanks are left alone.
  - Firmware changed the location of where it is storing the drive serial number.
  - Code was changed to retrieve the disk serial number from the new firmware location.
- Fixed issue where Windows delayed reporting the removal multiple drives simultaneously.
  - Seen by LSI lab. When an entire topology of disks was removed from the system, the Windows Device Manager did not display the change in topology in a timely manner.
  - Added call to Resume in execution of queued target resets.
- Fixed STP/SSP passthrough calculation of minimum input buffer size.
  - Seen in LSI lab.
- Fixed clean up of task management when issued via ioctl.
  - Seen in LSI lab. When issuing a target reset via an ioctl, the LSI Windows driver hangs.
  - Added code to clean up the reset state when a task management completes.

# Version 1.21.15

## Major Changes From Version 1.21.14:

### General Changes

#### Functionality

- None.

#### Defect fixes

- Added queuing of target resets.
  - Seen in LSI lab. With multiple drives including the boot drive attached to a HAB, and a stress test running to multiple non-boot drives, pulling the cable to the non-boot drives results in Windows hanging.
  - The driver was receiving target reset requests for all of the drives that were disconnected, but the driver was only sending down the first target reset it received.
  - Code was changed to queue target resets for later completion if a reset is already occurring.

# Version 1.21.14

## Major Changes From Version 1.21.13:

### General Changes

#### Functionality

- Added OEM SAS tape device recognition and enabling of TLR.

#### Defect fixes

- Added SAS AEN notification when a SMART error is detected and an event is added to the MPI event queue.

# Version 1.21.13

## Major Changes From Version 1.21.12:

### General Changes

#### Functionality

- Added ShutdownNotification to all INF files. Driver binary version change only.

#### Defect fixes

- None

# Version 1.21.12

## Major Changes From Version 1.21.11:

### General Changes

#### Functionality

- Added LSISAS1078 support.
- Added LSIFC949E support.
- Add SMART detection for all SAS devices, SEP message to light Predictive Fault LED for custom OEM SAS adapters only.

#### Defect fixes

- None

# Version 1.21.11

## Major Changes From Version 1.21.10:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed message alignment problem in SendIOCPAGE4 routine (buffer needs to be quadword aligned).
  - Seen in LSI lab. When JBOD with SCSI Enclosure Processor is attached, at boot time a bogus reply frame would be received with a context value of 0.
  - The request message buffer for a write of IOC Page 4 was improperly aligned on a dword boundary instead of the required quadword boundary.
  - Code was changed to check buffer alignment and realign on a quadword boundary if necessary.
- Change command to start DV process from Inquiry to Report Luns.
  - Seen by developer. Domain Validation actions were being triggered by receipt of an Inquiry command. However, StorPort issues a ReportLuns command before issuing an Inquiry. Domain Validation should be done on the first command to the device.
  - Fix is to change the trigger command to a Report Luns.
- Fix IOCTL timeout to complete IOCTL (dual timer call fix)
  - Seen by LSI lab. Reproduced by issuing IOCTLs to a FC device and pull device while IOCTL is active. IOCTL should timeout and be returned to the application with an error, but IOCTL was not being returned (and no subsequent IOCTL would be processed).
  - Root cause was determined to be StorPort behavior on RequestTimerCalls. A delayed timer routine is used to timeout an

IOCTL so cleanup can be done in a synchronized routine. However, if multiple timer calls are issued in one pass through the ISR, neither of the timer calls take effect.

- Fix is to track the RequestTimerCall requests and ensure that only one maximum is done in one ISR pass.



# Version 1.21.10

## Major Changes From Version 1.21.09:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed issue with QAS check config page messages. Needed to set DV\_CONTEXT\_FLAG so these won't be put back on the Message FIFO.
  - Seen in LSI test lab. System configuration with a mixed bus (both QAS and non-QAS drives) would hang during boot.
  - At the end of DV, if a mixed bus exists, SCSI Device Page 1 messages are sent to disable QAS on all drives that have it enabled. These messages didn't have the DV\_CONTEXT\_FLAG set so on completion entries were placed on the Message FIFO, causing corrupted context values.
  - Fix is to set the DV\_CONTEXT\_FLAG for these messages.
- Fixed GET\_DEVICE\_ADDRESS IOCTL (bug detecting IR phys disks). Mask off LogInfoValid bit when saving ConfigPageStatus.
  - Reported by OEM customer, reproduced by LSI developer.
  - IOCTL would return a SAS address for all target IDs after the last true existing target, instead of NoDeviceAddress error.
  - Fixed logic error in detecting IR physical disk and test of true IOCStatus for an invalid config page.

# Version 1.21.09

## Major Changes From Version 1.21.08:

### General Changes

#### Functionality

- None

#### Defect fixes

- Modified Domain Validation routines to fix synchronization problems during last ID (15).
  - Reported by LSI test lab.
  - Full duplex nature of StorPort driver caused a race condition for post-DV processing when DV the last ID (15) was completed. OS issued I/O would be completed twice, resulting in a bluescreen.
  - Fix is to use separate message frames for post-DV processing and wait until that processing is complete to complete the original I/O for ID 15.

# Version 1.21.08

## Major Changes From Version 1.21.07:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed GET\_SCSI\_BUS\_DATA (parallel SCSI) IOCTL (save Srb).
  - Reported by customer, reproduced by LSI.
  - Previous change introduced a bug where the current IOCTL Srb was not being saved, resulting in access of a bad pointer on IOCTL completion.
  - Fix is to save the IOCTL Srb at start of processing.
- Moved UpdatePersistencyTable from BringlocToOperationalState to HwInitialize (needs to run after EnableEvents). Also added deferred processing of UpdatePersistencyTable if ResetActive.
  - Reported by LSI test lab during error injection tests.
  - Pending events could block use of doorbell handshake.
  - If ResetActive doorbell could be in use, use timer routine to update the persistency table after reset has completed.
- Added option (via registry entry) to use a Link Reset in the place of a bus reset for FC (avoids dueling bus resets in multi-initiator configurations).
  - Reported by customer, reproduced by LSI. Applies to cluster configurations only.
  - Bus resets terminate all I/O's on a target for all initiators. This can result in another initiator timing out I/O's and issuing a bus reset, resulting in ping-ponging resets. A link reset terminates I/O's only for the issuing initiator.

- Added HWResetTimer for FC to avoid returning selection timeout if devices take a long time to log back in after controller reset
  - Reported by LSI test lab during error injection tests.
  - Some fabric controllers require too long for devices to log in, therefore the initially report only a subset of devices after a controller reset occurs. This results in DeviceNotThere returns until all devices have logged in.
  - Fix is to add a delay in returning DeviceNotThere status back to the OS until 30 seconds after a controller reset.

# Version 1.21.07

## Major Changes From Version 1.21.06:

### General Changes

#### Functionality

- Remove debouncing of WMI AEN's being sent (requested by RSA).
  - StorLib was missing events when debouncing was used, but gets all events when debouncing is not used.
- Added checks on LUN/target reset to fail operation if all outstanding I/O's in the scope are not completed. For bus reset added code to complete all outstanding I/O's on bus from driver tracking if not completed by F/W.
  - Changes were made to double-check termination of I/O's by the firmware.
- DelayedRestCompletion routine was removed (not needed due to reset changes above).

#### Defect fixes

- Add checks to CSMI GetRaidConfig IOCTL to not include a "foreign" (SAS vs. SATA) or "too small" hot spare drive as part of the volume.
  - Reported by OEM lab, reproduced by LSI.
  - To reproduce, create a SAS IR volume with hot spare, switch the hot spare drive with a SATA hot spare drive that is smaller than the IR volume physical disks. Run CSMITest GetRaidConfig IOCTL. Hot spare drive will appear in volume configuration.
  - Fix is to check hot spare drive type against volume drive type and check that hot spare size is adequate.
- Added clearing of TimerActive flag in ResetScsiTimer.
  - Detected by developer review of code.
  - If reset did complete, but ResetScsiTimer routine executed anyway, the WatchdogTimer would not be rescheduled.



# Version 1.21.06

## Major Changes From Version 1.21.05:

### General Changes

#### Functionality

- Added Phase III features to SAS\_PHY\_CONTROL IOCTL.

#### Defect fixes

- None

# Version 1.21.05

## Major Changes From Version 1.21.04:

### General Changes

#### Functionality

- Added logging of device add/remove for LogExceptionEvents.  
(requested for debug purposes)

#### Defect fixes

- Changed GUID for EVENT\_DATA\_IR in ca\_sas.mof and MEGASAS\_EVENT in ca\_wmi.c for StorLib WMI AEN support.
  - Found by RSA test lab. StorLib would get confused when both SCG and RSA SAS adapters were installed in a system, due to both drivers using the same GUID for the AEN Event structure.
  - Fix was to use a different GUID in the SYMMPI driver.
- Fixed input buffer length calculation in SSP and STP passthrough.
  - Found by LSI test lab. CSMLtest program uses an exact buffer size for SSP and STP tests. Calculation was off by 1 byte due to a U8 data[1] element at the end of the IOCTL structure that wasn't accounted for.
- Fixed GetDeviceAddress and GetLocation IOCTLs.
  - Found by OEM lab. IOCTLs were using IR physical drives as if OS could see them. IR physical drives should not be reported with an address or location of their bus/target is passed in as inputs. For GetLocation, if the IR volume bus/target is passed in, locations are returned for all IR physical disks.
- Changed EventAck message to use EventNotify reply frame. Increased number of reply frames to number of request frames plus 16 (FC only).
  - Found by LSI test lab. These changes were needed to handle extreme conditions when many I/O's are terminated and EventAck's get stuck behind the I/O's on FC.



- Fixed saving of IoctlSrb in SMP Passthrough.
  - Found in OEM lab, reproduced in LSI lab. SMP passthrough commands would timeout, causing delays in system boot.
  - Problem was routine saving Srb for comparison before an element was modified, so comparison failed on completion and IOCTL was not completed back to OS.
  - Fix was to save Srb after element was modified.

# Version 1.21.04

## Major Changes From Version 1.21.03:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed Diag Reset IOCTL crash (don't complete in PartitionMemory). Fixed IssueBusReset IOCTL (don't synchronize). Added reset of SCSI bus when IOCTL times out. Complete Diag Reset IOCTL after reset is fully completed.
  - Recent changes in IOCTL handing (queuing and miniport timing out of IOCTLs) required the above changes.
  - System would bluescreen when MPI\_DIAG\_RESET was issued due to double completion of IOCTL.
  - IssueBusReset IOCTL would hang due to multiple calls to SynchronizeAccess.
  - Changed MPI\_DIAG\_RESET to not complete until H/W reset was completed (eliminated need for utilities to add delays).
- Set PortEnable timeout to 300 seconds for SAS (IT and IR F/W).
  - Seen by OEM customer with large SAS topology. Host driver would time out PortEnable after 30 seconds.
  - Debugging showed that large SAS topologies (many expanders and PHYs) can take significantly longer than 30 seconds to do SAS discovery (and receive a reply from PortEnable). Increased the PortEnable timeout to 300 seconds for all SAS F/W types.

# Version 1.21.03

## Major Changes From Version 1.21.02:

### General Changes

#### Functionality

- For FC hard disk devices, force Inquiry data to support CmdQue, sync, and wide16 (some FC devices don't set these).

#### Defect fixes

- Modified SAS GetLocation IOCTL to check bus/target mapping and return error if not mapped.
  - Seen by LSI developer
  - Fix prevents request for SAS Device Page 0 for a non-existing page.
  - Return proper CSMI code of NO\_DEVICE\_ADDRESS.
- Fail LUN or Target reset if TerminationCount = 0.
  - Seen by LSI developer.
  - If LUN or Target reset completes with status of success but no I/O's were terminated, return error so StorPort will escalate to the next level of reset (prevents repeated attempts at LUN or target reset).
- Complete all outstanding I/O's after a bus reset.
  - Seen by LSI developer.
  - Prevents adapter or system hang if one or more I/O's is lost by the controller and not completed by the F/W.
- Modified check of RAID Phys Disk Change event to add a replaced physical disk SAS address to the cross-reference table.
  - Seen by OEM lab, reproduced by LSI lab.
  - Symptom: an IR physical drive which is replaced by a different physical drive will not display the new drive's SAS address in Get Raid Configuration and SSP Passthru's will not work.
  - Fix is to update the driver internal cross-reference table whenever a

RAID Phys Disk Status Change event with a disk status of on-line is received.

- Fixed synchronization problem with IOCTLs using tracking array.
  - Seen by customer OEM lab, reproduced in LSI lab.
  - When running agents (IOCTLs) and normal I/O's the system could run into hang conditions or bluescreens.
  - Issue was due to IOCTL queuing and starting of IOCTLs in the ISR thread which accessed tracking array pointers that are also accessed in the Startlo thread. This could result in overwriting of pointers or tracking entries.
  - Fix is to avoid access of tracking array for IOCTLs and using separate tracking mechanism.

# Version 1.21.02

## Major Changes From Version 1.21.01:

### General Changes

#### Functionality

- None

#### Defect fixes

- Added workaround for StorPortPause/Resume calls hanging when in crash dump mode.
  - Seen by LSI lab, reproduced by LSI developer
  - Hibernation and crash dump would not complete
  - Bug in Microsoft diskdump.sys driver, StorPortPause routine simply waits instead of pausing new I/Os.
  - Fix is to not call StorPortPause in crash dump mode (it is single-threaded anyway).
- Updated Domain Validation routine for StorPort synchronous operation (affects LSI\_SCSI.SYS only).
  - Seen in Microsoft lab, reproduced by LSI developer
  - Domain Validation routine was not properly synchronizing with StorPort full duplex mode of operation. System could hang during boot, Setup, or driver upgrade processes.
  - Modified Domain Validation routines to retain proper synchronization for full duplex operation.

# Version 1.21.01

## Major Changes From Version 1.20.20:

### General Changes

#### Functionality

- None

#### Defect fixes

- Added check for resync percentage reply with total blocks = 0.
  - Reported by OEM lab, reproduced by LSI lab
  - During IR hot plugging the RAID Action Indicator Struct reply can have Total Blocks = 0 (would cause a bluescreen due to a Divide-by-Zero error).
- Numerous changes for robust recovery if errors, hangs, faults occur while CSMI IOCTLS are active.
  - Reported by OEM lab and LSI lab
  - System hangs/bluescreens seen during IR hot plug testing.
  - Added multiple checks for outstanding IOCTLS during all error handling/recovery actions.
- Numerous synchronization changes to prevent double completions
  - Seen in Microsoft labs during LUN/target/bus reset stress testing (MS in-house test program).
  - Added code to ensure all reset processing is done synchronized.
- Added read of IOC Page 1 to get slotNumber in ReadAndSetSASPages.
  - Reported by OEM lab, reproduced by LSI developer
  - Slot number was always being reported as 0. Will now be value populated in IOC Page 1 by LSI SAS BIOS.
- Added setting of ResetActive and increased delay to 3 seconds on MPI\_DIAG\_RESET
  - Seen in LSI labs during adapter hard reset testing (Fibre Channel)
  - Changes align with error handling synchronization changes above.

- Added miniport queuing and timing out of IOCTL requests
  - Seen in OEM labs, reproduced in LSI labs. IOCTLs would hang with multiple threads running
  - StorPort does not single-thread IOCTLs (as ScsiPort does) nor does it time out IOCTLs (as ScsiPort does).
- Disabled quiescing/unquiescing of IR physical disks when sending a SSP or STP passthrough IOCTL.
  - When quiescing for each passthrough, IR F/W exhibits occasional anomalies in replies (no reply to SSP [IOCTL hang], incorrect RAID Action reply). Seen as IOCTL hangs.
  - Analysis determined that quiescing was not necessary.
- Fixed GetScsiAddress immediate return of Srb
  - Seen in OEM labs, reproduced by LSI developer
  - GetScsiAddress IOCTL call would hang.
  - Code changed to properly complete IOCTL back to port driver.
- Added code to populate SasAddressTable for new IR physical disk.
  - Reported by OEM lab, reproduced by LSI developer
  - If an IR physical disk is not present at system boot, then hot plugged later, a passthrough command to that disk will fail with an invalid parameter error.
  - SAS Address of the newly added IR disk was not in the driver's internal SAS address to SCSI bus/target cross-reference table.
  - Fix is to key off of the RAID IR Phys Disk Status Change event to populate the table.
- Fixed SSP Passthrough bug which truncated returned data.
  - Reported by OEM lab, reproduced by LSI developer
  - Issue seen on 64-bit systems as last 4 bytes of expected data were not returned.
  - Calculation for data length returned for entire IOCTL request was incorrect.

# Version 1.20.20

## Major Changes From Version 1.20.19:

### General Changes

#### Functionality

- Updated to MPI headers 1.05.10.
- Added ScsiMaxLunLimit registry entry to allow user to override 32 LUN limit for parallel SCSI.
- Added support for 1064E and 1066E.
- Added new PCI Config adapter info page and WMI AEN support for SAS (StorLib support).
- Added custom OEM IOCTLs Get Location, Get Connector Info, and Phy Control IOCTLs.

#### Defect fixes

- Added blocking of new I/O's while any Task Mgmt request is outstanding (per MPI specification).
- Removed StorPort CompleteRequest after LUN or Target Reset.
  - Could cause double completed commands, seen in Microsoft test labs.
- Added adapter Pause and Resume calls to block new I/O's during Task Mgmt requests (requested change by Microsoft StorPort developer).



# Version 1.20.19

## Major Changes From Version 1.20.18:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed issue that prevented the driver from supporting EEDP in T10 mode.

# Version 1.20.18

## Major Changes From Version 1.20.17:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed issue with U320 event driven DV (was due to an OS bus scan at the same time). Qualified RAID volume event processing to be done only for SAS devices.
  - Seen in LSI test labs. System would bluescreen on insertion of a RAID volume physical disk (secondary or hot spare).

# Version 1.20.17

## Major Changes From Version 1.02.15:

(Version 1.20.16 was skipped to stay in sync with the SYMMPI driver.)

## General Changes

### Functionality

- Added support for the SAS 8-port PCI-Express device (1068E).
- Modified driver enabling of EEDP (End-to-End Data Protection) to follow the Fibre Channel F/W setting in IOC Page 1.
- Added a BusType registry entry to all INF files (Windows 2000 and above).
- Added driver install package for XP x64 driver.

### Defect fixes

- Enhanced handling of error conditions during TM requests for LUN or Target Reset.
  - Seen in developer test system. Improved error recovery in those cases where F/W did not process TM requests successfully.
- Fixed a bug in calculating Max SG elements for SCSI IO 32 (was getting a data underflow due to 96-byte U320 message frame size). This affected U320 adapters only.
  - Detected in Microsoft labs, verified by LSI developer.
  - Calculations would underflow, causing the driver to provide ScsiPort with very large bogus numbers for NumberOfPhysicalBreaks and SrbExtensionSize.
  - Symptoms observed included bluescreens and hangs, but not on all systems.
- Added Task Mgmt Target Reset after a SAS device not responding event received (required by F/W to complete the removal of device).
  - Detected in LSI labs. In some cases, when a SAS or SATA drive was

hot removed, then another drive was plugged into the same PHY, the new drive was not detected by the F/W.

# Version 1.20.15

## Major Changes From Version 1.02.14:

### General Changes

#### Functionality

- Added PERSISTENCY\_TABLE\_FULL to HandleEventNotification to detect full persistency table for SAS devices.
- Added capability to issue CSMI SSP and STP passthrough I/Os to IR physical disks.
- Added handling of IR volume events to allow OS to detect volumes arriving or leaving.

#### Defect fixes

- Fixed Standby problem, set flag to prevent WatchdogTimer routine from resetting controller when coming out of Standby.
  - Seen in LSI test lab. Some systems would not resume when coming out of Standby.
- Fixed problem with config page accesses in HwInitialize due to events being enabled already (could read improper header info).
  - Seen in LSI test lab. If events were enabled before the reading of config pages during driver init, an event being received during the reading of config pages could result in improper data being read for a config page header or config page data.
- For IOC\_TERMINATED status and IOCLogInfo of aborted command, changed return status from BUSY to ERROR (OEM request, BUSY returns are retried by ScsiPort and aren't returned to the sending application).
- Fixed bug introduced when EEDP support was added (incorrectly building SG list for non-SCSI I/O messages).
  - Seen in developer testing. A non-SCSI I/O message that required a SG list would have that list built in the wrong place in the message.

# Version 1.20.14

## Major Changes From Version 1.02.13:

### General Changes

#### Functionality

- Added 939X/949X 4GB FC support.
- Added End-to-End Data Protection (EEDP) support for 949X/939X Fibre Channel devices.

#### Defect fixes

- In CheckMpioctlReply, set default ReturnCode to 0.
  - Found in LSI developer testing. The addition of SAS IOCTLs resulted in the SIC->ReturnCode being set to non-zero for generic MPI IOCTLs. This impacted utilities that use the generic MPI IOCTL.
- Added Hot Spare physical disks to GetRaidConfig IOCTL.
  - The list of physical disks in a RAID volume will now include all hot spare physical disks.

# Version 1.20.13

## Major Changes From Version 1.02.12:

### General Changes

#### Functionality

- For IOC\_TERMINATED status and IOCLogInfo of aborted SAS command, changed return status from ABORTED to BUSY (OEM request). Also, place IOCLogInfo value into Srb->TimeOutValue field to allow determination of reason for abort by test application.
- Added HWFaultLimit capability to catch SRAM fault codes and block adapter resets after user defined count of faults. (For FC SRAM checking, enabled via HWFaultLimit registry entry.)

#### Defect fixes

- Check to make sure we don't overwrite an active timer, when we fail to reset the SCSI bus (ResetScsiTimer)
  - Bug in Watchdog Timer code added in 1.20.10. Can result in adapter hang due to ResetActive not being cleared.
- Use 5 second timeout on config page requests via doorbell handshake (vs. 30)
  - Reduces error recovery time if doorbell handshake fails.
- When updating persistence table, quit if any handshake fails.
  - OEM testing saw an adapter hang due to handshake fail.

# Version 1.20.12

## Major Changes From Version 1.02.11:

### General Changes

#### Functionality

- Due to requirements from Microsoft, the LSIMPT.SYS driver has been split into 3 separate drivers based on protocol (LSI\_SCSI.SYS, LSI\_FC.SYS, and LSI\_SAS.SYS). These drivers are all built from the same source base, but have unique INF files to install only the adapters of a particular protocol for each driver.
- On F/W download boot performed during StopAdapter, don't wait for F/W to return to Ready state, leave immediately. Also, increased timeout in getting to Ready state in FindAdapter from 10 to 20 seconds.
  - This change was made to alleviate a Longhorn Watchdog timer assert in AdapterControl.

#### Defect fixes

- None



# Version 1.20.11

## Major Changes From Version 1.02.10:

### General Changes

#### Functionality

- None

#### Defect fixes

- Fixed hibernation and crash dump for SAS/U320 (was accessing FC config pages).
  - Found in LSI test lab. The addition of the FC WMI HBA API functionality (in version 1.20.10) required adding reads of FC config pages. In crash dump (and hibernate) mode, the driver defaults to FC chip type. The fix is to qualify the FC config page reads so they aren't done when in crash dump mode.

# Version 1.20.10

## Major Changes From Version 1.02.09:

### General Changes

#### Functionality

- Added a WatchdogTimer check for the firmware going non-operational (once per second). This will detect if the firmware faults must faster than waiting for an I/O timeout.
- Added support for WMI and HBAAPI for FC devices (required by Microsoft for all Fibre Channel drivers beginning with Server 2003 Service Pack 1).

#### Defect fixes

- Fixed UpdatePersistencyTable to try reading FCDevicePage0 using BUS\_TID before invalidating a currently-valid entry; if that is successful, then leave the entry valid.
  - For FC devices only. Problem seen in OEM and LSI test labs. Occasionally, FC devices would be unmapped if they did not login fast enough after a controller hard reset.
- Fixed SAS IOCTL code that was returning improperly on errors.
  - Seen in developer testing. For some IOCTLs, if an IOCTL returned an error, the SRB was being completed twice (once in the low level routine and again in an upper dispatch routine). This did not cause a problem since both were done on the same execution pass in the miniport.
- Always clear CpqTaskType in LSImpiReset.
  - Some I/Os terminated by an OS issued Bus Reset could be returned with an Abort status vs. a Bus Reset status.
- Fixed update of SasAddressTable on add/remove events.
  - Hot added SAS devices could not be accessed properly by some SAS IOCTL commands.

# Version 1.20.09

## Major Changes From Version 1.02.08:

### General Changes

#### Functionality

- None

#### Defect fixes

- Added a check of IOCLogInfo on SCSI\_IOC\_TERMINATED to determine whether an Abort or Reset was done, and return proper SrbStatus.

# Version 1.20.08

## Major Changes From Version 1.02.07:

### General Changes

#### Functionality

- Added RAID support to GetControllerConfig.
- Enabled QueryTask Task Management request for SAS IOCTLs.

#### Defect fixes

- Zero out redundant ROM fields in GetControllerConfig.
- Added setting of bDataPresent in NonSCB\_Func\_handler for SAS TM replies.
- Fixed copy of response data for SSP passthroughs in CheckMpiloctlReply.
- Fixed test for end device in GetScsiAddress.

# Version 1.20.07

## Major Changes From Version 1.02.06:

### General Changes

#### Functionality

- Updated to CSMI .83 header.
- Added support for new bSSPStatus field to task management and SSP Passthru.
- Add a SAS\_FW\_CACHE\_SIZE define and dynamically set the F/W cache size according to device type (SAS – 160K, others – 75K).

#### Defect fixes

- Change return status for CSMI clear task set and abort task set operations to SRB\_STATUS\_ABORTED.
  - Found in OEM testing. Injection of Bus Resets would result in I/O's being completed with Aborted status.
- Change the tracking mechanism for SRBs so that CSMI abort IOCTLs look for the SRB to abort in the IO tracking array. Also, include a member in the IO tracking array to store the IO path/target/lun/tag information so that the driver can match this information locally.
  - Found in OEM testing. Task Management Abort commands would only rarely be seen on the bus. Routine to determine if an I/O was still outstanding had a logic error that would not catch some outstanding I/Os.
- Fixed GetPhyInfo IOCTL to correctly get device type.
  - Code that set the PHY connected device type was using the field as a bitfield instead of an enumerated value.

# Version 1.20.06

## Major Changes From Version 1.02.05:

### General Changes

#### Functionality

- Added support for SAS 1068/1066 devices.
- Removed support for parallel SCSI (U320) devices from driver installation files (see Restrictions above)
- Enabled support for ClearTaskSet in custom OEM IOCTLs

#### Defect fixes

- Set CpqTaskType in custom OEM IOCTL to ensure that the return value for an aborted IO is correct.
- Changed INF device description strings (shortened slightly, EMC PowerPath display problem)

# Version 1.20.05

## Major Changes From Version 1.02.04:

### General Changes

#### Functionality

- None

#### Defect fixes

- Set PortEnable timeout according to F/W type (30 seconds for non-IR, 300 seconds for IR). This eliminates the 5 minute delay on non-IR F/W if there are errors during PortEnable.
- – Additional changes to allow completion of all outstanding I/O's after a hard reset to work. Fibre Channel testing exhibited some problems with completing all I/O's after a hard reset of the controller. These changes allow all I/O's to be completed successfully back to the OS.

# Version 1.20.04

## Major Changes From Version 1.02.03:

### General Changes

#### Functionality

- Added code to allow a Diagnostic Reset IOCTL to be executed even if the NoBoardReset flag is set.

#### Defect fixes

- Fixed custom OEM GetScsiBusData IOCTL to properly handle 64-bit data buffer address.
- Modified LsiMpiReset to not count unanswered resets (use timer instead)
  - With high numbers of outstanding I/O's, multiple resets are issued by the port driver. The unanswered reset count was going above the threshold, causing a hard reset of the port. On dual-channel adapters, this could cause ping-ponging resets.
  - Observed in LSI lab, reproduced by issuing a hard reset with LsiUtil under I/O load on a dual-channel adapter.
  - Fix is to remove the unanswered reset count. This was added long ago to trap failed F/W. A timer has already been added which will detect failed F/W and will reset the port.
- Enable doorbell interrupts for all devices except 919X/929X
  - Previously, doorbell interrupts were disabled for all Fibre Channel devices. However, the 919X and 929X devices are the only ones with a H/W errata that can cause a PCI PERR if the IntStatus register is read.
  - Driver modified to use doorbell interrupts except for 919X/929X.
- Change generic MPI IOCTL to issue Task Management requests through the doorbell.
  - Task Management requests must be issued through the doorbell.
  - This change allows utilities to issue any type of Task Management request through the generic MPI IOCTL I/F.



- Fixed OEM custom GetDeviceData IOCTL to properly report U320 device speed.
  - Driver was reporting a device speed of async for U320 devices.
  - Table index in searching for speed value was incorrect.
- Set ReplySize and ConfigReplySpace to size of IOCFactsReply.
  - Size was hard coded to 64 bytes, then in 1.20.03 it was raised to 80 bytes for Fibre Channel and SAS devices.
  - To enable compatibility with future MPI changes to IOCFacts, the size for FC and SAS is now determined by the size of the IOCFactsReply in the MPI headers used to build the driver.
  - The size for 1020/1030 will remain 64 bytes for compatibility with older applications that have this size hard coded.
- Fixed re-enable of interrupts for device reset.
  - Doorbell interrupts were being enabled for 919X/929X devices when they should not be used due to H/W errata.
  - Could result in a system hang after device resets have been issued (due to doorbell interrupts not being serviced by ISR).
- Fixed processing of SASDeviceChangeStatus event to notify OS to scan the bus when a device is removed.
  - Observed in customer lab, OS detection of a removed SAS/SATA device did not occur when the device was removed, only when another device was installed.
  - Driver now issues a BusChangeDetected notification on both a removal and arrival of a SAS/SATA device.
- Added code to clean up all outstanding I/O's when the device is hard reset.
  - Observed hang condition with StorPort driver, long delays with ScsiPort driver. Reproduced by issuing a hard reset via LsiUtil while under I/O load.
  - If the port needs to be hard reset, outstanding I/O's will not be completed back to the OS, resulting in hangs or long timeouts.
  - Driver will now complete all outstanding I/O's back to the OS with a SCSI bus reset status 1 second after a hard reset has been done.

# Version 1.20.03

## Major Changes From Version 1.02.02:

### General Changes

#### Functionality

- Added Serial Attached SCSI (SAS) device support.
- Added support for version MPI 1.5 interface.
- Added support for InBand Management.
- Added support for Firmware Diagnostic Buffers.
- Added driver based Fibre Channel bus/target persistency (registry enabled).
- Added support for custom OEM SAS IOCTLs.
- Added an additional OEM SSVID to enable setting the Initiator ID for U320 adapters at 255 (required for support of OEM SAF-TE device).

#### Defect fixes

- Added support for auto request sense disabled (happens during crash dump).
- Changed default size of MPI reply frames to be 64 for all U320 devices and 80 for Fibre Channel and SAS devices.
  - Problem observed in LSI labs at in OEM customer testing.
  - Some system management applications hard coded the reply frame size to be 64 bytes. These versions are used only with U320 devices.
  - SAS devices require a reply frame size of 80 bytes due to the increased size of the IOCFacts reply.

# Version 1.20.02

## Major Changes From Version 1.09.17:

(Version label bumped to 1.20.02 as due to starting the addition of SAS device support.)

## General Changes

### Functionality

- Added U320 tape workaround (tape detect and IDP bit). (Needed for tape drive support at U320 speeds.)
- Added SAS device support (still in-progress, not enabled in INF).

### Defect fixes

- Modified 919X/929X MOST workaround for rev. 0 only.
  - MOST workaround not required for 929XL.
- Bumped the default reply frame size (and static reply buffer size from 64 to 80 bytes).
  - Necessary to get full IOCFacts reply frame due to increased size for SAS devices.
- Fixed check of outstanding TM request.
  - Problem exhibited as Fibre Channel adapter not supporting crash dump, due to trying to access a SCSI config page.
- Changed MPIOMode handling to work with multi-path filter drivers in both multiple path and single path modes.
  - Observed in LSI labs, Fibre Channel adapters in a multi-pathing configuration.
  - Failure mode: If paths have been degraded such that there is only a single path available to a device, then LIPs are done, the testing application will receive I/O errors due to devices being marked as missing.
  - Fix: For non-Microsoft MPIO multi-pathing drivers, do not enable the extended FCP status processing. This processing would return a

Selection Timeout status due to the FCP status return on a LIP. This is MPIOMode = 1. For the Microsoft MPIO driver, this Selection Timeout status is necessary, and MPIOMode = 2 should be used to enable FCP status.

# Version 1.09.17

## Major Changes From Version 1.09.12:

### General Changes

#### Functionality

- Added NumberOfRequestMessageBuffers registry entry to allow the maximum number of outstanding commands to be limited.
  - Allows user to lower the maximum number of outstanding I/Os for testing purposes, or to avoid hitting device queue limits.
- Removed support for FC909 adapter (obsolete, end-of-life).

#### Defect fixes

- Numerous bug fixes to IOCTL processing in error handling situations.
  - IOCTLs could be completed twice during error handling, resulting in system hangs or bluescreens.
  - System configuration: injecting error conditions (cable breaks) while running I/O stress tests and storage management agents.
- Fixed problem with Event Ack reply (uses static message frame).
  - Request message frame was being returned to the message FIFO, even though it was actually a statically allocated frame.
  - System would bluescreen if an event was received that required an Event Ack message to be sent.
- Modified EnableDiagnosticRegister routine to always write a "junk" byte before the magic sequence. (1020A stayed in the DiagReg enabled mode after writing the H/W reset bit.)
  - Systems with a flashless 1020A would fail Windows power management (system hang).
- Added timer to Task Management Reset reply. If no reply is received within 15 seconds, the F/W is assumed to be dead, a H/W reset is done and the TM request is retried.
  - Seen on flashless 1030 systems under Microsoft PnP Stress test.
  - I/O's would stop on failed adapter.



# NOTES, ISSUES and DETAILS

## 1. Known issues and restrictions

- The BIOS Config Utility settings for sync, wide, and adapter SCSI ID are the only ones honored by the Windows driver. Other settings are overridden by the Windows OS.
- IOCTL calls cannot be issued to an adapter that has no visible devices on the bus (Windows OS restriction).
- IOCTL calls are not supported on Windows 9X\_ME.
- For full driver upgrade functionality in flashless environments, Windows drivers at revision 1.08.12 and above and 1030 F/W at revision 1.00.14.00 or above should be used.
- LSI Pseudo Device support is enabled (via a registry entry) by default on Fibre Channel devices, starting with version 1.09.09. (Not enabled for Server 2003. A system registry entry provides the same function.)
- When using 1030 IR F/W version 1.03.09 or later, driver version 1.09.05 or later must be used for hibernation and crash dump to be supported properly.
- Driver versions from 1.09.07 through 1.09.10 have the PCI-X Multiple Outstanding Split Transactions setting in config space set to 1 for 919X and 929X. This can result in data corruption. Fixed in 1.09.11.
- Driver version 1.09.15 is required for proper operation with MyStorage.
- Driver version 1.09.90 or above is required for 1020A flashless environments.
- Multiple adapters of the same type should all have the adapter BIOS enabled if one of the adapters controls the boot device.
- To enable End-to-End Data Protection (EEDP) support for the 949X/939X devices, the registry entry "EEDP\_T10\_Enable=1;" must be added to the DriverParameter string.

- Beginning with version 1.20.17, Serial Attached SCSI (SAS) device support for Server 2003 is supplied only by the LSI\_SAS.SYS StorPort-based driver.
- For the Windows 2000 and Windows Server 2003 operating systems a hotfix is required for proper support of drive hot plug operations. This is described in Knowledge Base article 867818. More information can be found at:  
<http://support.microsoft.com/default.aspx?scid=kb;en-us;867818>
- Versions of LSI\_SCSI, LSI\_SAS, and LSI\_FC at or before 1.24.03 can cause memory corruption under a very specific system environment. See the “Defects fixes” section under version 1.24.04

Disclaimer:

LSI Logic Corporation and/or its respective suppliers make no representations about the suitability of the information contained in these documents and related graphics. All such documents and related graphics are provided “AS IS” without warranty of any kind. LSI Logic Corporation and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all implied warranties and conditions of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall LSI Logic Corporation and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information contained in these documents and related graphics. Some states do not allow the exclusion or limitation of liability for consequential damages so the foregoing limitation may not apply.