

Release Notes
Microsemi Adaptec Host Bus Adapter 1000
Series

Released
August 2020



a  **MICROCHIP** company

Revision History

Revision	Revision Date	Details of Change
17	August 2020	Updated for 1.9 release.
16	August 2019	Updated for 1.8 maintenance release.
15	March 2019	Updated for 1.7.1 patch release.
14	January 2019	Updated for 1.7 maintenance release.
13	September 2018	Updated 1.6.1 patch release.
12	April 2018	Updated for 1.6 maintenance release.
11	October 2017	Updated for 1.5 maintenance release.
10	June 2017	Updated for 1.4 maintenance release.
9	January 2017	Updated for maintenance release.
8	November 2016	Updated for November maintenance release.
7	September 2016	Updated for 9/15 maintenance release.
6	April 2016	Updated for maintenance release.
5	December 2015	Updated for production release.
4	November 2015	Update for pre-production release.
3	September 2015	Update for pre-production release.
2	August 2015	Update for pre-production release.
1	August 2015	Document created.

Software License Agreement

PLEASE READ CAREFULLY: THE USE OF THIS SOFTWARE IS SUBJECT TO THE SOFTWARE LICENSE TERMS OF MICROSEMI, INC. AND OTHER LICENSORS WHOSE SOFTWARE MAY BE BUNDLED WITH THIS PRODUCT.

BY YOUR USE OF THE SOFTWARE INCLUDED WITH THIS PRODUCT YOU AGREE TO THE LICENSE TERMS REQUIRED BY THE LICENSOR OF THAT SOFTWARE, AS SET FORTH DURING THE INSTALLATION PROCESS. IF YOU DO NOT AGREE TO THE LICENSE TERMS APPLICABLE TO THE SOFTWARE, YOU MAY RETURN THE ENTIRE UNUSED PRODUCT FOR A FULL REFUND.

In return for acquiring a license to use the Microsemi software, which may include software from third party licensors and patches made available by Microsemi (“Software”), and the related documentation, you agree to the following terms and conditions:

1. License. This Agreement grants you, the Licensee, a license to:
 - a. Use the Software on a single computer system, which is not intended for use by more than five (5) users; and:
 - b. Make one copy of the Software in machine readable form solely for back-up purposes, provided you reproduce Microsemi's copyright proprietary legends. Notwithstanding the foregoing, the Software may be used on the home, laptop or other secondary computer of the principal user of the Software, and an additional copy of the Software may be made to support such use. As used in this license, the Software is “in use” when it is either loaded into RAM or installed on a hard disk or other permanent memory device. The Software may be “in use” on only one computer at any given time. (Different license terms and fees are applicable for networked or multiple user applications.) As a specific condition of this license, you agree to use the Software in compliance with all applicable laws, including copyright laws, and that you will not copy, transmit, perform or distribute any audio or other content using the Software without obtaining all necessary licenses or permissions from the owner of the content.
2. Restrictions. You may not distribute copies of the Software to others or electronically transfer the Software from one computer to another over a network. You may not post or otherwise make available the Software, or any portion thereof, in any form, on the Internet. You may not use the Software in a computer service business, including in time sharing applications. The Software contains trade secrets and, in order to protect them, you may not decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form. YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, LOAN, RESELL FOR PROFIT, DISTRIBUTE, NETWORK OR CREATE DERIVATIVE WORKS BASED UPON THE SOFTWARE OR ANY PART THEREOF.
3. Ownership of Software. As Licensee, you own the media upon which the software is recorded or fixed, but Microsemi and its licensors retain title and ownership of the Software recorded on the original media and all subsequent copies of the Software, regardless of the form or media in which or on which the original and other copies may exist. This license is not a sale of the Software or any copy.
4. Confidentiality. You agree to maintain the Software in confidence and that you will not disclose the Software to any third party without the express written consent of Microsemi. You further agree to take all reasonable precautions to preclude access of unauthorized persons to the Software.
5. Term. This license is effective until January 1, 2045, unless terminated earlier. You may terminate the license at any time by destroying the Software (including the related documentation) together with all copies or modifications in any form. Microsemi will have the right to terminate our license immediately if you fail to comply with any term or condition of this Agreement. Upon any termination, including termination by you, you must destroy the Software (including the related documentation), together with all copies or modifications in any form.
6. Special Terms Applicable to Databases. Where a database is included with the Software, you acknowledge that it is licensed only in connection with the use of the Software to perform disc creation, and that the

database and all data derived therefrom must be maintained in confidence in accordance with the provisions of Section 4. This license does not grant you any rights to distribute or disclose such database or data.

- 7. Limited Warranty.** Microsemi and its Licensor warrant only that the media upon which the Software is furnished will be free from defects in material or workmanship under normal use and service for a period of thirty (30) days from the date of delivery to you. MICROSEMI AND ITS LICENSORS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. THE FOREGOING STATES THE SOLE AND EXCLUSIVE REMEDIES MICROSEMI AND ITS LICENSORS WILL PROVIDE FOR BREACH OF WARRANTY. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, MICROSEMI AND ITS LICENSORS MAKE NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED, AS TO NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.
- 8. The entire liability of Microsemi and its licensors, and your exclusive remedy for a breach of this warranty, shall be:**

 - a.** The replacement of any media not meeting the above limited warranty which is returned to Microsemi; or:
 - b.** if Microsemi or its distributor is unable to deliver replacement media which is free from defects in materials or workmanship, you may terminate this Agreement by returning the Software and your money will be refunded.
- 9. Limitation of Liability.** IN NO EVENT WILL MICROSEMI OR ITS LICENSORS BE LIABLE TO YOU FOR ANY INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR LOSS OF DATA, EVEN IF MICROSEMI OR A LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Some states do not allow the exclusion or limitation of special, incidental, or consequential damages, so the above limitation or exclusion may not apply to you.
- 10. Export.** You acknowledge that the laws and regulations of the United States and other countries may restrict the export and re-export of the Software. You agree that you will not export or re-export the Software or documentation in any form in violation of applicable United States and foreign law.
- 11. Government Restricted Rights.** The Software is subject to restricted rights as follows. If the Software is acquired under the terms of a GSA contract: use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract. If the Software is acquired under the terms of a DoD or civilian agency contract, use, duplication or disclosure by the Government is subject to the restrictions of this Agreement in accordance with 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors and 49 C.F.R. 227.7202-1 of the DoD FAR Supplement and its successors.
- 12. General.** You acknowledge that you have read this Agreement, understand it, and that by using the Software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between Microsemi and you, and supersedes any proposal or prior agreement, oral or written, and any other communication between Microsemi and you relating to the subject matter of this Agreement. No additional or any different terms will be enforceable against Microsemi unless Microsemi gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of Microsemi. You assume full responsibility for the use of the Software and agree to use the Software legally and responsibly. This Agreement shall be governed by California law, except as to copyright matters, which are covered by Federal law. This Agreement is deemed entered into at Sunnyvale, California by both parties. Should any provision of this Agreement be declared unenforceable in any jurisdiction, then such provision shall be deemed severable from this Agreement and shall not affect the remainder hereof. All rights in the Software not specifically granted in this Agreement are reserved by Microsemi.

Should you have any questions concerning this license, contact:



Microsemi Corporation
Corporate Headquarters
One Enterprise
Aliso Viejo, CA 92656 USA

Contents

1 About This Release.....	1
1.1 Release Identification.....	1
1.2 Release Components.....	2
1.2.1 Firmware and UEFI/HII and Legacy (CTRL-A) BIOS.....	2
1.2.2 Drivers.....	2
1.2.3 Host Management Software.....	2
1.2.4 Supporting Documents.....	3
2 What is New?.....	4
2.1 Features.....	4
2.2 Fixes.....	8
2.2.1 Firmware Fixes.....	8
2.2.2 UEFI/BIOS Fixes.....	17
2.2.3 Driver Fixes.....	25
2.2.4 Management Software Fixes.....	39
2.3 Limitations.....	45
2.3.1 Firmware Limitations.....	45
2.3.2 BIOS/UEFI Limitations.....	47
2.3.3 Driver Limitations.....	48
2.3.4 Management Software Limitations.....	52
3 Implementing a Host Application for a Bus Management Controller.....	54

1 About This Release

This document describes the production release package of supported firmware, OS drivers, and host management software files for Microsemi® Adaptec® HBA 1000 Series products: 1000-16i, 1000-16e, 1000-8i8e, 1000-8i, 1000-8e.

Note: All Microsemi Adaptec products are UL listed and for use only with UL listed ITE.

1.1 Release Identification

The firmware, software, and driver versions for this maintenance release are shown in [Table 1 • Release Summary](#) on page 1.

Table 1 • Release Summary

Solutions Release	SR 1.9
Package Release Date	August 14, 2020
Firmware Version	4.90b0 (basecode 05.01.030.002) ¹
UEFI Version	1.2.6.3
Legacy BIOS Version	1.2.6.4
Microsemi Adaptec RAID Controller Configuration Utilities (ARCCONF Command Line Interface, maxView Storage Manager, maxView vSphere Plugin, Boot USB, EventMonitor)	3.02.00.23699
Drivers Package Version	B59002

Note:

1. This release requires that, if not already done, you first update older boards to firmware version 1.00 build 0 *before* updating to this firmware release. See issue 3 of this document for important notes with regards to updating older firmware to version 1.00 build 0.

1.2 Release Components

Download the following firmware, drivers, host management software, and supporting documentation for your HBA 1000 Series product from the Microsemi Adaptec Web site at <https://start.microsemi.com/>.

1.2.1 Firmware and UEFI/HII and Legacy (CTRL-A) BIOS

This release supports version 4.90 build 0 of the HBA 1000 firmware, version 1.2.6.3 of the Microsemi Adaptec UEFI/HII, and version 1.2.6.4 of the Legacy (CTRL-A) BIOS utilities.

1.2.2 Drivers

Drivers for this release have been tested on the following operating systems. Drivers can be loaded on out-of-box operating system versions, the latest service pack, or software updates. Compatibility issues may be seen with untested OS versions.

Microsoft Windows Certified

- Microsoft® Windows® Server 2019 RS5/LTSC
- Microsoft® Windows® Server 2016 RS1/LTSC
- Microsoft® Windows® Server 2012 R2 SP1, 2012 SP1 (64-bit)
- Microsoft® Windows® Server 2008 R2 SP1 (64-bit)
- Microsoft® Windows® 10 (1909) (64-bit)
- Microsoft® Windows® 8.1, 8 (64-bit)
- Microsoft® Windows® 7 SP1 (64-bit)

Linux for x64

- Red Hat® Enterprise Linux/CentOS 8.1, 8.0 (64-bit)
- Red Hat® Enterprise Linux/CentOS 7.7, 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, 7.0 (64-bit)
- Red Hat® Enterprise Linux/CentOS 6.10, 6.9, 6.8, 6.7, 6.5 (64-bit)
- SuSE Linux Enterprise Server 15 SP1, 15 (64-bit)
- SuSE Linux Enterprise Server 12 SP5, SP4, SP3 (64-bit)
- Ubuntu Linux 18.04.3, 18.04.2, 18.04.1, 18.04 (LTS; 64-bit)
- Ubuntu Linux 16.04.5 (LTS) (64-bit)
- Oracle Linux 7.7 with UEK 5 (64-bit)
- Oracle Linux 6.9 with UEK 3 and UEK 4 (64-bit)

Note: Additional driver binaries are provided in the release package, but are provided as-is without support.

Citrix

- Citrix XenServer 8.1, 8.0, 7.6 LTS, 7.5 LTS, 7.4 LTS, 7.3 LTS, 7.2 LTS, 7.1 LTS (64-bit)

1.2.3 Host Management Software

This release of the HBA 1000 Series provides Microsemi Adaptec Controller Configuration Utilities for host management, which include the following:

- Microsemi Adaptec ARCCONF Command Line Utility
 - Windows (64-bit)
 - Linux (64-bit)
 - VMware EXSi 6.5 and 6.0 (64-bit)
 - XenServer 7.0 (64-bit)

- Microsemi Adaptec maxView Storage Manager
 - Windows (64-bit)
 - Linux (64-bit)
 - VMware EXSi 6.5 and 6.0 (64-bit)
 - XenServer 7.0 (64-bit)
- Microsemi Adaptec maxView vSphere Plugin
 - vCenter 6.5 and 6.0
- Boot USB (offline or pre-boot) for the ARCCONF command line utility and maxView Storage Manager
 - Linux (64-bit)
- Virtual OS Environments
 - Citrix XenServer 7.0 (64-bit)
 - Microsoft Hyper-V (64-bit)

Note:

Hyper-V runs as a service on the Windows host OS and does not require a separate driver.

Refer to the *README: maxView Storage Manager and ARCCONF Command Line Utility* for specific OS support details, installation notes, and known issues.

1.2.4 Supporting Documents

Refer to the Microsemi Adaptec Web site at <https://start.microsemi.com/> for the latest product documentation for your HBA 1000 Series product.

PDF

- *Host Bus Adapter 1000 Series Installation and User's Guide*, PMC-2152188.
- *Host Bus Adapter 1000 Series Command Line Utility User's Guide*, PMC-2152156.
- *maxView Storage Manager User's Guide for Microsemi Adaptec Smart-Family Controllers*, PMC-2153109.
- *README: maxView Storage Manager and ARCCONF Command Line Utility*, PMC-2153191.
- *Host Bus Adapter 1000 Series Release Notes* (This Document), PMC-2152169.

HTML

- *maxView Storage Manager Online Help*

2 What is New?

2.1 Features

[Table 2 • Feature Summary](#) on page 4 lists features supported for this release. Features new to this release are designated as "New".

Table 2 • Feature Summary

Feature		Supported in this Release	Future Release
Software Features			
Support for 0 based direct attached drive numbering		X	
Support for up to 256 ^[1] SAS/SATA target devices		X	
Multi-LUN support		X	
Concurrent RAID and HBA support		X	
SATA NCQ HDD		X	
Storage space support	Expander attach	X	
Persistent binding	SCSI ID of a device/LUN remains the same after reboot - No addition or removal of devices	X	
Hot Plug drive support		X	
Backplane port/box/bay support with no backplane management		X	
SATA drive/SSD support		X	
HDD/SSD Work Load logging			X
SSD Wear Gauge			X
>2 TB physical drives		X	
S.M.A.R.T.		X	
SAS Drive/SSD support		X	
SAS expander support	Expose enclosure to host	X	
	SMP pass through to expander	X	
External storage 6G/12G		X	
External Array Connect		X	
Managed Cables		X	

Feature		Supported in this Release	Future Release
Clustering/HA support for SAS devices (Windows and Linux)			X
Mixed mode port configuration (full support)		X	
RAID Features [3]			
RAID	RAID 0, 1, 10, 5	X	
	Up to 32 drives per volume	X	
	Up to 8 volumes	X	
	Split Mirror and Combine (duplicate configuration by mirroring)	X	
	Copy back hot spare	X	
	Heal array (replace drive or restart rebuild)	X	
	Move array (move all data from one array to another)	X	
	Erase drive (erase physical drive's data)	X	
	Logical drive rebuild	X	
	Staggered drive spin up for direct attached	X	
	Configurable stripe size	X	
	Dynamic sector repair	X	
	Bad stripe marking	X	
Hot Spare support	Dedicated (activated for a failed data drive for any assigned array)	X	
	Auto Replace (activated for a failed data drive for a single assigned array)	X	
	Predictive (replaces a drive after it reports a Smart error with the spare)	X	
Build modes	Quick initialization	X	
	Rapid parity initialization	X	
	Background initialization	X	
Dual Path Failover	RAID stacked managed	X	
	OS managed for HBA mode (MPIO)	X	
Microsoft Storage Spaces/HA certification		X	

Feature	Supported in this Release	Future Release
Tape Support		
Internal Tape 3 GB SAS LTO3/4 I	X	
Internal Tape 6 GB SAS LTO5/6	X	
External Tape 3 GB Multi-LUN, Tape Library	X	
External Tape 6 GB Multi-LUN, Tape Library	X	
Flashing		
Recovery NOR Flash ROM image	X	
Offline controller firmware flash	X	
Online controller firmware flash with reboot	X	
Offline drive firmware flash	X	
Online drive firmware flash with reboot	X	
Tape drive flashing	X	
Firmware recovery utility for an inoperable controller (DOS-based)	X	
Logging		
Device and array error logs	X	
Detailed controller log through serial port	X	
Detailed controller log through log files	X	
BMC Support		
BMC DST mechanism consistent with Series 8	X	
Enclosure Management		
SES-2	X	
SES-3	X	
SGPIO (SFF-8485)	X	
IBPI (SFF-8489)	X	
Management Utilities		
Microsemi Adaptec UEFI/HII BIOS Configuration Utility	>2.2 TB boot	X
	UEFI boot from legacy device/RAID array	X

Feature		Supported in this Release	Future Release
	UEFI boot from raw device (HBA device)	X	
	HII GUI (config support)	X	
	Secure boot	X	
Microsemi Adaptec Legacy BIOS Configuration Utility ^[3]	Legacy BIOS boot from legacy device/RAID array	X	
	Legacy BIOS boot from raw device (HBA device)	X	
	Legacy configuration at boot (CTRL-A)	X	
Microsemi Adaptec Controller Configuration Utilities	ARCCONF Command Line Interface (Windows, Linux, VMware)	X	
	maxView Storage Manager	X	
	Boot USB	X	

Note:

1. 238 SSD/HDDs maximum support and remainder are reserved for expanders and enclosure management.
2. RAID logical volumes are not supported with 4kn devices.
3. Only UEFI systems can boot from a 4K sector drive. The legacy CTRL-A controller BIOS displays all 4K and 512 byte drives connected to the controller during POST and there is CTRL-A support (like Format and Verify) for both 4K- and 512-byte drives, but INT 13 support for booting is available for 512-byte drives only.

2.2 Fixes

2.2.1 Firmware Fixes

2.2.1.1 Fixes and Enhancements for Firmware Release 4.90 Build 0

This release includes the following fixes and enhancements.

- Fixed an issue during drive hot replacement where fault LED turns ON and continues to be ON even after the drive was replaced
 - *Root cause:* When drive gets hot removed, there is a small window where individual drive presence status bit does not get cleared, leaving the fault LED to get turned ON incorrectly.
 - *Fix:* Drive presence status bit field is now set/cleared correctly during hot removal/add cases.
 - *Risk:* Low

2.2.1.2 Fixes and Enhancements for Firmware Release 4.80 Build 0

This release includes the following fixes and enhancements.

- Fixed a controller lockup when logging events to host interface.
 - *Root cause:* A logical IO can result in a double completion when logging host events. The second completion results in a lockup because the resource is already freed.
 - *Fix:* Clear appropriate pointers so a logical IO double completion will not occur.
 - *Risk:* Low
- Fixed an issue where the box index and bay number are the same for a direct-attached drive and drive behind the enclosure on the same connector.
 - *Root cause:* For drives on the same connector, the box number is initialized to the same value.
 - *Fix:* Reordered direct-attached and enclosure initialization code, so direct-attached drives/enclosure drives on the same connector get different box numbers.
 - *Risk:* Low
- Fixed an issue where an I2C bus would hang with a bad drive attached.
 - *Root cause:* There are multiple slave devices in one specific server that are connected to the same TWI bus in their design. When the controller goes into a lock-up state, interrupts are disabled and the basecode does not respond to slave requests. Because of this, the TWI master cannot no longer reach the other TWI slave devices on the bus.
 - *Fix:* Create a new HAL API that allows the RAID Stack to disable any TWI slave port that it manages on lockup.
 - *Risk:* Low

2.2.1.3 Fixes and Enhancements for Firmware Release 4.52 Build 0

This release provides the following fixes and enhancements:

- Fixed an issue where Western Digital drives would not accept any IOs after running the Sanitize Crypto Scramble command.
 - *Root cause:* The problem occurred only when using Western Digital drives. After the Sanitize Crypto Scramble had completed, the drive would not accept any IOs until about 9 minutes later. The `hdparm -sanitize-status` commands report the Sanitize Crypto Scramble operation is complete; however, sending `sg_turs (TEST UNIT READY)` commands to the drive shows that the firmware was returning LOGICAL UNIT NOT READY, SANITIZE IN PROGRESS. The firmware was out of sync with the state of the drive.

- *Fix:* If firmware has set the "Sanitize in progress" flag for the target device, send Sanitize Status Ext in the background for any commands sent by host.
- *Risk:* Moderate

2.2.1.4 Fixes and Enhancements for Firmware Release 4.50 Build 0

This release provides the following fixes and enhancements:

- Fixed a controller lockup issue when a tape drive is connected.
 - *Root cause:* Lockup occurred when firmware used incorrect logical request resources across tape-related thread.
 - *Fix:* Pass the logical request resource, its buffer, and buffer length as parameters to caller, so it is used correctly across threads.
 - *Risk:* Low

2.2.1.5 Fixes and Enhancements for Firmware Release 4.02 Build 0

This release provides the following fixes and enhancements:

- Integrated Microsoft-signed UEFI driver.
- Improved controller boot time by parallelizing the drive discovery process instead of processing one drive after another.
- Fixed an issue where the controller could lock up during boot when processing faulty SEP devices.
- Fixed an issue where the physical drives part of a RAID volume was exposed to the host OS by properly setting the Peripheral Qualifier PQ bit in Inquiry data.
- Fixed a blank screen display on the CTRL+D menu option in the CTRL-A utility that lists the array members. The fix is to correctly handle the global variable related to the screen display.
- Fixed an issue where a controller deadlock could occur during an HBA check condition if there were no available event buffers to report the check condition. This is fixed by using callback functions instead of event buffer mechanism.
- Fixed an issue during controller reset recovery where the outstanding host commands could get processed by firmware, just after the driver invalidates the host firmware interface causing a lockup. The fix is to not process the outstanding commands, as the controller is going to go through a reset and the host will resubmit the command.
- Fixed an issue where the activity LED will be turned on whenever there is IO to any drive and get updated every 1 second.
- Fixed an issue where after deleting a RAID volume, the firmware was not sending a host event to the driver to rescan, causing stale file nodes to be present on host OS.
- Fixed an issue where the `Smartctl` toolset was unable to get the HDD smartinfo in RAID mode. This is now enabled in firmware by allowing read-only SCSI commands to the HDD drives portion of the logical device.
- Fixed enclosure errors messages ("ENCLOSURE ERROR: encl/encl_sgpio.c LINE 922") in controller serial log when a drive is hot-plugged in the SGPIO bay. This is fixed by properly reordering related function calls.

2.2.1.6 Fixes and Enhancements for Firmware Release 3.52 Build 0

This release provides the following fixes and enhancements:

- Added support for logging critical events like drive hot add, hot remove, and physical drive failure events on PBSI interface; PBSI DST, version is incremented to 1.5.
- Added support for Chenbro enclosure types 80H10343208 and 80H10343207.

- Added support for `smp utils` commands from the host by exposing the expander to host in controller HBA mode.
- Added IOBypass support for 4Kn drives when using HBA mode.
- Updated controller serial output buffer logs to use the system time after the OS driver has provided it.
- Optimized the external SAS PHY settings for the HBA 1000 for SAS speeds 6G, 3G, and 1.5G. These settings are updated through the SEEPROM file update.
- Added improvements to the controller lockup/crash functionality to collect more critical information.
- Added support to log an entry to the trace log when failing a drive.
- Added support to send IT nexus resets during command timeout handling for SEP devices instead of sending device resets to SEPs.
- Added support to use the direct-parity-write optimization during volume transformations.
- Added functionality to update drive LEDs that are moved out of critical section.
- Added support for additional error thresholds and handling for SATA drive timeout cases.
 - *Root cause:* Some bad drives would often time out repeatedly, resulting in a loop of continuous drive resets.
 - *Fix:* Added support to detect such drives and, if they experience two resets within a two-minute window, the drive will be marked offline (which can be brought back online using host tools).
 - *Risk:* Low
- Fixed an issue where UEFI/CTRLA hangs during boot when the enclosure is powered off during a small time window.
 - *Root cause:* During device discovery, if the enclosure's discovery fails, a stale firmware error status of a reused physical request is returned, causing a virtual SEP device discovery failure and a UEFI/CTRLA hang at the host.
 - *Fix:* For the SES device's internal commands, reset the statuses of the physical requests firmware status to "good".
 - *Risk:* Low
- Fixed an issue where UEFI HII allows the creation of logical devices with invalid IOBypass options.
 - *Root cause:* The host tools show the IOBypass option for logical devices.
 - *Fix:* Turned off the IOBypass firmware feature that exported invalid options to the host tools; this enables the host tools to only present valid options.
 - *Risk:* Low
- Fixed an issue where the controller would lockup due to double completion of a single IO.
 - *Root cause:* During hot addition of drives, there was a corner case where the same internal IO resource can complete twice by two different threads
 - *Fix:* Changed the secondary thread to use the proper IO resource rather than reusing the IO resource used by another thread.
 - *Risk:* Low
- Fixed an issue where the rebuild of RAID 1 array was stuck at 0% after a driver unload/reload.
 - *Root cause:* When the driver is unloaded, the background disk tasks are paused explicitly; however, it was not resumed automatically during driver reload.
 - *Fix:* Resume the background disk tasks if it was paused earlier during a driver unload operation.
 - *Risk:* Low
- Fixed an issue where firmware background activities like PHY error check will stop if the OS driver is unloaded and reloaded without a system reboot.
- Fixed an issue where no host events are generated when there are no active consumers like maxView.
 - *Root cause:* When there are no active event consumers, an unwanted consumer overflow event is generated causing the firmware to stop generating events.

- *Fix:* Consumer overflow event is not generated when there are no active event consumers.
- *Risk:* Low
- Fixed an issue where background disk task was not starting PBSI Service when no drives were connected.
 - *Root cause:* The PBSI DST table initialization bit was not set when no drives were attached, causing the background disk task to not start PBSI Service.
 - *Fix:* PBSI service is now started regardless of whether disk is present.
 - *Risk:* Low
- Fixed an issue where the controller would lockup 0xEA when power cycling JBOD within a short span of time.
 - *Root cause:* When the enclosure is power-cycled continuously within short span of time, an event overflow would cause the controller to send out-of-sync diagnostic commands to the JBOD, resulting in lockup.
 - *Fix:* Added logic to detect and consolidate the event overflow, which helps in sending the diagnostic commands in time, thus preventing lockup.
 - *Risk:* Low
- Fixed an issue where a fewer drives were detected during the enclosure hot-plug test in a multiple-enclosure configuration.
 - *Root cause:* The controller was not able to detect a few expander PHY connector types, causing the drives to be dropped.
 - *Fix:* The expander PHY connector type is now identified and all drives are added properly during the enclosure hot-plug event.
 - *Risk:* Low
- Fixed an issue where after an expander firmware update, the prior expander firmware version was still displayed on host tools.
 - *Root cause:* After a non-disruptive expander firmware upgrade, the old version was being displayed by the controller firmware.
 - *Fix:* After an expander non-disruptive firmware update, recheck the expander in order to get the latest expander firmware version so that it can be exported to host.
 - *Risk:* Low
- Fixed an issue to fill proper connector name for non-SES boxes.
 - *Root cause:* The firmware connector name is not being filled out for non-SES boxes.
 - *Fix:* The firmware will fill out the proper connector name for non-SES boxes.
- Fixed an issue where resetting all controller ports does not work from host tools.
 - *Root cause:* When you need to reset all ports in the controller, host tools issue a command with SCSI ID LOW and HIGH set to 0xFF, respectively. Both of them are a byte wide and, the firmware checks MAX_DWORD after calculating the device ID from the given SCSI ID. Due to this, the firmware reports the device 65535 as invalid.
 - *Fix:* Compare SCSI ID LOW and HIGH to 0xFF to ensure all ports are reset.
- Fixed an issue where hot plugging a drive on a backplane with sgpio backplane and internal expander connected causes the drives to nest improperly in SSA.
 - *Root cause:* sgpio logic was clearing out all box information if sdatain was not available.
 - *Fix:* Do not clear out box information for internal expanders.
 - *Risk:* Low
- Fixed an issue where additional internal commands are sent to a failed drive invoking unwanted additional error handling mechanisms to be executed.
 - *Root cause:* No explicit checks are done for failing drives.
 - *Fix:* Added additional checks so further internal commands are not sent to a failing drive.

- Fixed an issue where hot-plugging a predictive failed drive caused the new drive to appear predictive failed.
 - *Root cause:* Background was checking stale Monitoring and Performance (MP) data once the drive was spun up, then setting the drive to predictive failure.
 - *Fix:* Clear the MP data in pre-spinup after you know the serial number.
 - *Risk:* Low
- Fixed an issue where the controller locked up while updating drive bay information on an enclosure with failed SEP.
 - *Root cause:* There was a race condition where a SEP was failed while a drive was hot added.
 - *Fix:* Added further checks to validate the SEP state.
 - *Risk:* Low
- Fixed an issue where the Identify LED executed from BIOS could cause BIOS hang and the backplane might not be operational after that point.
 - *Root cause:* Firmware reused the data from the SES Status page, modifying the data to fit the SES Control Page. This caused a reserved bit to have a 1, causing SEP to fail.
 - *Fix:* Clear the reserved bits in the SES Control Page before sending.

2.2.1.7 Fixes and Enhancements for Firmware Release 3.02 Build 0

This release provides the following fixes and enhancements:

- Fixed an issue where the user was unable to select a boot device if a secondary boot device was already selected.
 - *Root cause:* Memory corruption.
 - *Fix:* Freed up unused code
 - *Risk:* Low
- The Locate LED behavior has been changed in this 3.02 firmware release. Previously, when the Locate LED is enabled, if a drive is removed and replaced the Locate LED will remain enabled and require the SSA/Arconf/maxView tools to be used to disable the LED. In this release, when the Locate LED is enabled, if a drive is removed and replaced the firmware will automatically disable the Locate LED so the host tools will not be required.
- Fixed an issue where firmware may intermittently hang during boot.
 - *Fix:* The firmware will perform the CPLD initialization only if the CPLD Type is set to a non-zero value in the SEEPROM.
 - *Risk:* Low
- Fixed an issue where the Enclosure LED was not working as expected in RAID mode when SEP was exposed to host.
 - *Fix:* If connector mode is non-HBA, allow the SES LED control page to update independent of the Host SEP exposure settings.
 - *Risk:* Low
- Fixed an issue where the tape backup was not working.
 - *Root cause:* A few fields in the firmware response structure to the driver were not set correctly.
 - *Fix:* The firmware will correctly set the auto sense valid field and response structure size field so that the host driver can pass the SCSI sense data buffer to the OS/tape application.
 - *Risk:* Low
- Fixed an issue where the Host BMC command timed out (set/clear boot drive).
 - *Root cause:* When writing the SEEPROM, interrupts are disabled. If a large block of data is written, other I2C interrupts may not get processed and cause timeouts.

- *Fix:* The SEEPROM will now be written in smaller blocks so that I2C interrupts are not disabled for long periods of time and command timeouts do not occur.
- *Risk:* Low
- Enabled IOBypass as the default option to preboot (BIOS/UEFI) utilities.
- Fixed an issue with the external HDD Status LED during and after a sanitize operation.
- Fixed expander off-by-one LEDs in dual domain configuration.
 - *Root cause:* For a given enclosure, we check all paths using both IO modules and their associated expanders. During these checks, we re-initialize some of the values. If good values for element index and eiioe bit were found on the first pass, on the second pass we were reinitializing over the good first values. If the second pass doesn't contain data, then they will not be updated with valid data.
 - *Fix:* Only re-initialize values once for any number of loops.
 - *Risk:* Low
- Reenabled IOBypass for volumes after rapid parity initialization (RPI) completed for volume.
 - *Root cause:* When the volume would begin RPI, the firmware disabled IOBypass for all units in the array.
 - *Fix:* At the end of RPI complete unit, recheck IOBypass for all units.
 - *Risk:* Low

2.2.1.8 Fixes and Enhancements for Firmware Release 2.56 Build 0

This release provides the following fixes and enhancements:

- Fixed an issue where after certain JBODs were reset, the controller firmware was not able to correctly identify the device bay numbers during SMP discovery.
 - *Root cause:* The controller firmware was not able to obtain the device bay numbers during initial part of the discovery, as some of the attributes were not available to calculate the bay numbers correctly.
 - *Fix:* Changed the controller firmware to populate the device bay numbers when the other attributes needed for calculating the bay number are available later in the discovery process.
- Fixed an issue where the SES discovery code was resetting the content of the SES elements when the controller was reset, thus resulting in drive bays being powered on.
 - *Root cause:* The firmware was resetting all the SES page 0x2 array element control bits during controller initialization.
 - *Fix:* Removed the code that was clearing the SES elements during initialization and preserving the page 0x2 contents across the controller firmware resets when the controller is in HBA mode.
- Fixed an issue where the firmware was responding to the pass through command that is sent to a non-existent (SGPIO device) target by the management software running in the background. This resulted in the firmware logs getting filled up with pass through failure messages, rendering the logs less readable.
 - *Root cause:* The firmware should check for the validity of the target before passing the command to the layers below.
 - *Fix:* The firmware added valid target check to the SCSI pass through command path and rejected the command if it was sent to the non-existent target.

2.2.1.9 Fixes and Enhancements for Firmware Release 2.50 Build 0

This release provides the following fixes and enhancements:

- Reduced the size of the basecode log buffer to match the size of the crash dump buffer in order to ensure that all entries of the basecode logs are stored in non-volatile RAM and are available for crash dump analysis.
- Fixed an issue where SEPs were exposed when only physical devices were exposed to the host.
 - *Root cause:* SEPs were exposed when only physical devices were exposed to the host.
 - *Fix:* When a configuration is created, SEPs are hidden from the host. There is a configurable option that can be set through set controller parameters that will set SEPs to be visible to the host at all times.
 - *Exposure:* All previous versions.
 - *Risk:* Low
- Fixed an issue where the Drive Ready LED was set to be off by default.
 - *Root cause:* Drive Ready LED was set to be off by default.
 - *Fix:* Customers can modify the default Drive Ready LED behavior by modifying the SEEPROM value. New bits are defined in the Tweak Settings section. See *Board Customization* document for additional details.
 - *Exposure:* This is a new configurable behavior. If a customer needs to change the SAS drive activity LED behavior, then they can do so through SEEPROM settings.
 - *Risk:* Low
- Added additional customer specific Sub-System IDs (SSIDs).
 - *Root cause:* Fixed an issue where firmware would assert when customer specific SSIDs were added.
 - *Risk:* Low
- Fixed an issue where PBSI was disabling after updating the firmware.
 - *Root cause:* On the first boot after updating, the default PBSI configuration must be initialized based on the default settings in the SEEPROM. After setting the PBSI configuration, the RAIDstack continued to use the original settings read at boot (which are all zeros). Since the clock speed is 0, PBSI is disabled for this boot.
 - *Fix:* After initializing the user PBSI settings, the PBSI settings need to be re-read before initializing the PBSI interface.
 - *Exposure:* Firmware release 2.00b0 and later.
 - *Risk:* Low
- Fixed an issue where the controller issued an ABORT TASK command only 20 seconds after the Start Unit.
 - *Root cause:* The internal command logic was ignoring the flags set and only explicitly allowing certain commands to set their timeouts.
 - *Fix:* Allow Start Unit to set its timeout through internal command
 - *Exposure:* All previous versions.
 - *Risk:* Low
- Fixed an issue where the controller would assert when a 0-byte CSMI SMP passthrough command was sent to a device.
 - *Root cause:* The firmware SMP protocol layer did not support sending 0-byte transfers to a device and would lock up.
 - *Fix:* Added additional checks in the CSMI SMP passthrough layer to fail any passthrough command that has a 0-byte request length before trying to send it down to the device. In addition, the firmware now checks to make sure the CDB length is also valid.
 - *Exposure:* All previous versions.
 - *Risk:* Low
- Fixed an issue where the controller would hang when simultaneously creating a spare drive in background while failing another drive from the error task.

- *Root cause:* The error task holds the big spinlock while calling in to save the configuration data. If the drive has not spun up yet, then saving the configuration data will attempt to restart the drive, calling `OS_sleep` while the big spinlock is held.
- *Fix:* : Put pause/resume critical section around `OS_sleep` in `RequestDriveSpinUpOrDown`, which will temporarily end the critical section so that there is no hang.
- *Exposure:* All previous versions.
- *Risk:* Low
- Added support to return the target port descriptor as part of INQUIRY to device identification VPD page (83h).
 - *Root cause:* Storage spaces were looking for the target port descriptor be returned by SATL as part of INQUIRY to device identification VPD page (83h). Looking at both SPC5 and SAT4, this descriptor is optional and was therefore not being returned.
 - *Fix:* Return target port descriptor as part of INQUIRY to device identification VPD page (83h).
 - *Exposure:* All previous versions.
 - *Risk:* Low
- Fixed an issue where non-configured drive activity LEDs were on when there was no activity.
 - *Root cause:* The behavior was changed accidentally in release 1.50b0 and is now being reverted back.
 - *Fix:* For non-configured drives, make the activity LED state off when the drives are inactive.
 - *Exposure:* Firmware release 1.50b0 and later.
 - *Risk:* Low
- Fixed an issue where the controller Virtual SEP (VSEP) was not exposed under Linux when the controller was in RAID mode.
 - *Root cause:* With the new Linux driver, the host does not scan all the buses and the VSEP is not getting exposed.
 - *Fix:* The firmware now exposes the virtual SEP in RAID mode.
 - *Exposure:* This issue will only be seen when driver build 53005 and newer are used with firmware prior to 2.50b0.
 - *Risk:* Low
- Fixed an issue where direct attached devices were starting at slot#1 instead of slot#0.
 - *Root cause:* Slot numbering was starting at 1 according to default SEEPROM settings.
 - *Fix:* The SEEPROM option was modified to start the slot numbering from 0. When the internal slot number starts at 0, the external numbering is also adjusted.
 - *Exposure:* All previous versions. Issue is present across all utilities (CTRL-A, PMCSSACLI, ARCCONF).
 - *Risk:* Low
- Changed the controller firmware behavior to continue the system startup despite a SAS scan failure (due to all drives not being spun up).
 - *Root cause:* If discovery fails for direct attached drives, the firmware will return a discovery failed status. When this happens, the firmware will abandon the topology resulting in no direct attached drives being listed. In this case, SATA drives remain in a spin up hold state and do not present the controller with any identification information. This results in a discovery timeout which causes discovery to fail.
 - *Fix:* The firmware will provide additional status when discovery fails. In the case of drives being stuck in the spin up hold state, a status value reflecting this will be returned. This value is used to display the drives that have been successfully discovered.
 - *Exposure:* All firmware versions will have this issue when there is a drive stuck in spin up hold (SUH).
 - *Risk:* Low

2.2.1.10 Fixes and Enhancements for Firmware Release 2.04 Build 0

This release provides the following fixes and enhancements:

- Added tweak value in the platform customizable image to use 0-based SGPIO bay numbering (default is 1-based).
- Added support for firmware to re-query discovery parameters of a drive such as firmware version.
- Added an `async` host event for firmware update.
- Added BMC TWI configuration settings to the board configuration area of SEEPROM.
- Added additional physical drive, logical drive, and physical connector parameters to PBSI interface.
- Added support for PBSI feature requests for the following:
 - (1) Controller mode
 - (3) Stripe size
 - (4) PFA indication for PDs
 - (6) PD state (rebuild and erasing)
- Added support to soft-fail 4Kn drive targets using the SSACLI "failpd" parameter.
- Fixed an issue where the drive temperature could be reported as 0 °C during boot.
- Modified drive temperature polling frequency to 60 seconds.
- Fixed an issue where CSMI SSP pass-through requests completed with incorrect status.
- Fixed an issue where CSMI SSP pass-through commands would not receive sense data.
- Fixed an issue with incorrect JBOD bay numbering during hot-plug.
- Fixed an issue with repeated drive spin-ups on spun-down spares related to temperature monitoring.
- Fixed an issue where the accelerated I/O queuing path was not managed properly for disks in Mixed Connector mode.
- Fixed an issue where a multi-LUN tape device is not correctly exposed to the host.
- Fixed a controller lockup while checking SES content of a hot-added JBOD.
- Fixed a controller lockup during boot when using "Rapid Parity Initialization".
- Fixed several issues with invalid cabling checks for wide port configurations.
- Fixed a controller lockup during hot-plug of a tape device.
- Fixed an issue where a SATA SSD could be incorrectly flagged for "SSD Wearout" after also logging predictive failure.
- Enhanced the handling of parity errors. This includes updated logging mechanisms and processing of parity exceptions.
- Fixed an issue where SSDs could not be accessed after an SSD firmware update.
- Fixed an issue with CSMI SMP pass through when called from the out-of-band management interface.
- Fixed an issue when iozone test tool and the threadx memory pool is near exhausted.

2.2.1.11 Fixes and Enhancements for Firmware Release 1.54 Build 0

This release provides the following fixes and enhancements:

- Fixed a controller lockup when a wrong SCSI error response data length of zero is returned to a BMC if a pass-through SCSI request resulted in error.
- Fixed an issue related to HDD's reporting "maximum" and "threshold" temperature values when those values are not supported by the drive.
- Fixed a controller lockup that could occur as a result of hot-adding an expander or drives to an internal connector.
- Fixed an issue where the connector name information in a dual domain configuration was not correct for the API returning enclosure status.
- Fixed an issue where the amber LED would fail to be lit if a drive was failed during discovery.

2.2.2 UEFI/BIOS Fixes

2.2.2.1 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.6.3/1.2.6.4

There are no changes for UEFI/HII Build 1.2.6.3 from the previous release.

For the Legacy BIOS, this release provides the following fixes and enhancements:

- Fixed an issue where an Incorrect Array name would show when the array count exceeds 26 in Legacy BIOS.
 - *Root cause:* Only one of the two characters were being used to display the array name.
 - *Fix:* Use both of the characters while printing the array name.
 - *Risk:* Low

2.2.2.2 Fixes and Enhancements for UEFI and Legacy BIOS Build 1.2.6.3

This release provides the following fixes and enhancements:

- Fixed an issue in CTRL-A utility where changing the spare activation mode was not allowed after deleting the auto replace spare drive on RAID0 logical drive.
 - *Root cause:* Even though the spare drives are deleted from the array, the auto replace flag will still be set for which the check returns that the spare mode cannot be changed as there is an auto replace spare.
 - *Fix:* Check the auto replace flag if and only if there are spare drives present in the array.
 - *Risk:* Low
- Fixed an issue in HII where user is not able to set drive write cache for unconfigured physical drives.
 - *Root cause:* Incorrect value passed to set drive write cache.
 - *Fix:* Value corrected in submit method for setting drive write cache.
 - *Risk:* Low

2.2.2.3 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.5.2/1.2.5.4

There are no changes for UEFI Build 1.2.5.2 from the previous release [Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.5.2/1.2.5.2](#) on page 18

There are no changes for Legacy BIOS 1.2.5.4 from the previous release [Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.5.2/1.2.5.2](#) on page 18

For the Legacy BIOS Build 1.2.5.4, this release provides the following fixes and enhancements:

- Boot device entry not cleared from CTRL A when boot device has been used for Array creation
 - *Root cause:* When a bootable device is configured, the device is only cleared from the bootable device entry when the user scrolls to the page where the drive is present. Otherwise, the device continues to be displayed.
 - *Fix:* When the user enters the Set Boot Device page, check whether any devices are configured. If a device is configured, remove the device from the bootable drive entry.
 - *Risk:* Low
- Fixed an issue where the SSD I/O Bypass Acceleration option was available in the controller pre-boot tools.
 - *Root cause:* The SSD I/O Bypass Acceleration option was available using the Ctrl+A utility in the Array-level help text.
 - *Fix:* The SSD I/O Bypass Acceleration option is disabled.
 - *Risk:* Low

2.2.2.4 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.5.2/1.2.5.2

For the UEFI/HII, this release provides the following fixes and enhancements:

- Fixed an issue where several options in the Controller Settings menu could not be modified.
 - *Root cause:* The setup browser does not consider the opcode menu items, which shared the same question key value. It expects to have a unique QID for every opcode.
 - *Fix:* Every opcode item to have a unique question key value.
 - *Risk:* Low

There are no changes for Legacy BIOS Build 1.2.5.2 from the previous release [Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.5.1/1.2.5.2](#) on page 18

2.2.2.5 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.5.1/1.2.5.2

For the UEFI/HII, this release provides the following fixes and enhancements:

- Fixed an issue where the 8K stripe size option is shown for logical drive creation in HII whereas the option is hidden in other tools.
 - *Root cause:* For logical drive creation, the 8K stripe size option has to be removed in HII to maintain consistency with other tools.
 - *Fix:* Removed 8K stripe size option.
 - *Risk:* Low
- Fixed an issue where the system would hang during boot on an OEM platform vendor.
 - *Root cause:* The return status for navigating the SMBIOS table was handled improperly.
 - *Fix:* Properly handled the return status to navigate the SMBIOS table.
 - *Risk:* Low
- Fixed an issue where the connector information was not valid in the Driver Health message.
 - *Root cause:* The phys_connector variable returns only the character 'N' and the number.
 - *Fix:* Connector string obtained from Long Connector Name instead of phys_connector.
 - *Risk:* Low
- Fixed an issue where controllers with different firmware versions were inaccessible in the BIOS setup menu.
 - *Root cause:* Two drivers claiming driver binding supported for same device due to incorrect logic in handling different driver versions.
 - *Fix:* Appropriate logic added to handle different driver versions. Highest version driver to handle all the devices.
 - *Risk:* Low
 - *Exposure:* All previous versions.
- Fixed an issue where the help text and warning information were not available in the Save Support Archive menu.
 - *Root cause:* Help text and warning information missing in Save Support Archive menu
 - *Fix:* Help text and warning information added in Save Support Archive menu.
 - *Risk:* Low
 - *Exposure:* All previous versions.
- Fixed an issue where the Controller Driver Health/HII Menu Name and Help text was not applicable.
 - *Root cause:* Need proper Controller Driver Health/HII Menu Name and Help text.
 - *Fix:* Modified Controller Driver Health/HII Menu Name and Help text to applicable "Controller Name".

- *Risk*: All previous versions.
- *Exposure*: Low
- Fixed an issue where the PCI slot number was not displayed for a few platforms.
 - *Root cause*: Several platforms do not populate the SMBIOS table prior to the execution of the EFI driver; as a result, the driver does not get the PCI slot number.
 - *Fix*: Fetch the PCI slot number using EFI Legacy BIOS platform protocol if the driver does not get it from SMBIOS protocol method.
 - *Risk*: Low
 - *Exposure*: All previous versions.
- Fixed an issue where the wrong string was displayed when a spare drive was changed from a dedicated spare to an auto-replace spare and vice versa.
 - *Root cause*: Improper title strings used for spare menu.
 - *Fix*: Title strings updated depending on the type of spare.
 - *Risk*: Low
 - *Exposure*: All previous versions.
- Fixed an issue where the controller firmware serial logs and diagnostic report were missing in the Save Support Archive option.
 - *Root cause*: The controller firmware serial log and diagnostic report support not enabled for Save Support Archive option.
 - *Fix*: Controller firmware serial log and diagnostic report support enabled for Save Support Archive option.
 - *Risk*: Low
 - *Exposure*: All previous versions.
- Fixed an issue where the disk firmware update failed through EFI HII.
 - *Root cause*: Improper CDB value was used when doing a disk firmware update.
 - *Fix*: Corrected the CDB value used for disk firmware update.
 - *Risk*: Low
 - *Exposure*: All previous versions.

For the Legacy BIOS, this release provides the following fixes and enhancements:

There are no changes for the Legacy BIOS Build 1.2.5.2 from the previous release.

- Fixed an issue where the wrong string was displayed in logical drive details in the CTRL-A utility when CTRL+S is pressed to check spare drive details.
 - *Root cause*: The condition check for the spare drive type was incorrectly implemented.
 - *Fix*: Rewrote the condition check for detecting the spare drive type.
 - *Risk*: Low
- Fixed an issue where a spare drive shared with multiple arrays was changing from a dedicated spare to an auto-replace spare in one of the arrays.
 - *Root cause*: The existing spare drives were not checked to verify if they were shared across multiple arrays.
 - *Fix*: Checked if the existing spare drive is shared across multiple arrays. If its shared, do not show the auto-replace option in the spare type selection menu.
 - *Risk*: Low
- Fixed an issue where the incorrect available size was displayed when creating multiple logical drives in the same array.
 - *Root cause*: A global variable was not reset to 0.
 - *Fix*: For every new array creation, initialized the global variable to 0.

- *Risk: Low*
- Fixed an issue where a physical drive set as a secondary bootable device took precedence over a logical drive set as the primary bootable device when the controller is in Mixed Mode.
 - *Root cause:* The Legacy BIOS enumerates the bootable physical drives first and the logical drives are enumerated later. Since the physical drive is already set as bootable, it takes precedence over the logical drive, which is set as primary.
 - *Fix:* Swapped the primary bootable logical drive and the secondary bootable drive in the above scenario.
 - *Risk: Low*
- Fixed an issue where you were unable to create more than 23 logical drives on an 8 TB drive in the CTRL-A utility.
 - *Root cause:* If the size exceeded more than 2 TiB, a 64-bit variable was not used to specify the size of the logical drive.
 - *Fix:* If the size exceeds 2 TiB, use a 64-bit variable to specify the size of the new logical drive to be created.
 - *Risk: Low*

2.2.2.6 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.4.1252/1.2.4.1254

For the UEFI/HII, this release provides the following fixes and enhancements:

- Fixed an issue where the minimum stripe size for logical drive creation is shown as 8 KB instead of 16 KB.
 - *Root cause:* The minimum stripe size assigned was 8 KB.
 - *Fix:* Changed minimum stripe size to 16 KB.
 - *Risk: Low*
 - *Exposure:* All previous versions
- Fixed an issue where a UEFI driver boot hang was seen on a vendor platform.
 - *Root cause:* Boot hangs while processing EDK helper EFI_ERROR method.
 - *Fix:* EFI_ERROR method converted to check EFI_SUCCESS status.
 - *Risk: Low*
 - *Exposure:* All previous versions
- Fixed an issue where the Driver Health messages had the wrong connector information.
 - *Root cause:* Truncated characters were used to display connector information.
 - *Fix:* Use Long Connector Name instead of truncated characters.
 - *Risk: Low*
 - *Exposure:* All previous versions
- Added support to clear the MBR/GPT partition while deleting the logical drive.
 - *Root cause:* New enhancement.
 - *Fix:* The signatures for the MBR or the GPT partition were mangled.
 - *Risk: Low*
 - *Exposure:* All previous versions
- Updated Help and Prompt Strings.
 - *Root cause:* Needed more enhanced strings.
 - *Fix:* Updated the strings to convey more detailed information to the user.
 - *Risk: Low*
 - *Exposure:* All previous versions

- Fixed an issue where the Windows Server 2008 R2 operating system was not booting into OS when partitions were created for all raw devices.
 - *Root cause:* SCSI READ/WRITE 16 commands in the BlockIO failed.
 - *Fix:* Use SCSI READ/WRITE 10 if READ/WRITE 16 is unsupported.
 - *Risk:* Low
 - *Exposure:* All previous versions
- Fixed an issue where Windows 2012 R2 was not booting up when it was installed in the first logical drive after the creation of multiple logical drives and reboot.
 - *Root cause:* The last block address for Block IO protocol was incorrect for all logical drives except the first logical drive.
 - *Fix:* Corrected logic to set proper last block address.
 - *Risk:* Low
 - *Exposure:* All previous versions
- Fixed an issue where the controller slot number was shown as 255
 - *Root cause:* A few servers do not populate the PCI slot information during the driver execution phase.
 - *Fix:* At the Exit Boot services phase, PCI slot information was getting populated and the same is reported to the firmware.
 - *Risk:* Low
 - *Exposure:* All previous versions

For the Legacy BIOS, this release provides the following fixes and enhancements:

- Fixed an issue where the 8 KiB stripe was listed for array creation.
 - *Root cause:* Unsupported stripe size.
 - *Fix:* Removed the unsupported stripe size.
 - *Risk:* Low
 - *Exposure:* SmartIOC 2000 controllers
- Fixed an issue where the array name was not correct when a physical drive was assigned as a hot spare in Legacy BIOS.
 - *Root cause:* The physical drive was not checked against the spare drive details.
 - *Fix:* The drive is checked against the data drive details and the spare drive details.
 - *Risk:* Low
 - *Exposure:* SmartIOC 2000 controllers
- Fixed display issues in Legacy BIOS.
 - *Root cause:* Outdated copyrights version and unreadable character displayed in the POST.
 - *Fix:* Updated the copyrights and removed the unreadable character.
 - *Risk:* Low
 - *Exposure:* HBA 1000 Series adapters
- Fixed an issue where the BMC setting was unavailable with HBA1000 SEEPROM.
 - *Root cause:* The Controller Settings menu was hidden for pure HBA controllers.
 - *Fix:* The Controller Settings menu is now enabled for HBA controllers.
 - *Risk:* Low
 - *Exposure:* HBA 1000 Controllers
- Fixed an issue where RAID 5/10 was displayed as RAID-Unknown in Boot Override Menu in setup utility.
 - *Root cause:* Comparison of unequal size operands always returned false.
 - *Fix:* Modified the size of the operands for the comparison.

- *Risk*: Low
- *Exposure*: SmartIOC 2000 controllers
- Fixed an issue where the array name was changed when the logical drive was set as a boot drive.
 - *Root cause*: The bootable logical drive was sorted as the first logical drive so the associated array name was also being changed.
 - *Fix*: Removed the sorting of the logical drives.
 - *Risk*: Low
 - *Exposure*: SmartIOC 2000 controllers
- Fixed an issue where the system was unable to save the queue depth value if only the queue depth value was modified.
 - *Root cause*: Save changes option for only queue depth change was not added.
 - *Fix*: Save changes option was added if only queue depth is changed.
 - *Risk*: Low
 - *Exposure*: SmartIOC 2000 controllers
- Fixed an issue where the failed logical drives were displayed in the Boot Override menu.
 - *Root cause*: The failed logical drives were added to the bootable device list.
 - *Fix*: Added conditions to prevent the failed logical drive from getting added to the bootable device list.
 - *Risk*: Low
 - *Exposure*: SmartIOC 2000 Controllers.
- Add warning message for SECURE ERASE option.
 - *Root cause*: No warning message was displayed to the users before starting the secure erase operation.
 - *Fix*: Added new warning message before starting the secure erase operation.
 - *Risk*: Low
 - *Exposure*: SmartIOC 2000 Controllers.

2.2.2.7 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.3.1202/1.2.3.1206

There are no changes for UEFI Build 1.2.3.1202 from the previous release [Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.3.1202/1.2.3.1201](#) on page 22.

For the Legacy BIOS, this release provides the following fixes and enhancements:

- Fixed an issue where the bay number in disk utilities was listing only two digits.
- Changed the BIOS controller boot message from "Waiting for controller to start..." to "Controller firmware starting & scanning for devices..."

2.2.2.8 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.3.1202/1.2.3.1201

For the UEFI/HII, this release provides the following fixes and enhancements:

- Fixed an issue where creating an offset logical drive providing 0 GiB as size succeeds with a random size logical drive.
 - *Root cause*: Error in size validation for 0.
 - *Exposure*: Create Logical Drive under Manage Array menu.
 - *Fix*: Improved size validation method for corner cases.
- Fixed an issue where the device type of the backup array was set to "Split mirror set backup orphan" after a split mirror operation.

- *Root cause:* Utility creates logical drives with the same signature for all the logical drives created. OS utility fails signature check to identify parent logical drive.
- *Exposure:* Create Array and Create Logical Drive menus.
- *Fix:* Utility will now generate and assign unique signature for each logical drive based on time stamp and unit ID.
- Fixed an issue where the Delete Spare fails if offset logical drives that have RAID0 and RAID1 as types are present.
 - *Root cause:* Spare map was ignored if RAID0 was present.
 - *Exposure:* Manage Spare menu.
 - *Fix:* Modifications to find other RAID types to get spare map.
- Fixed an issue where Utility hangs when Delete Spare operation done on an array that has 50 logical drives.
 - *Root cause:* Physical Drive information, which is required for listing, was not handled properly.
 - *Exposure:* Delete Spare operation.
 - *Fix:* Modified Physical Drive Information gathering to be handled properly.
- Added a validation check for duplicate logical drive names.
 - *Root cause:* No error message was displayed if duplicate logical drive name is used during offset logical drive creation.
 - *Exposure:* Create Array menu.
 - *Fix:* Validation check added for duplicate logical drive names.
- Fixed an issue where logical drive creation was not being blocked during logical drive transformation and migration.
 - *Root cause:* Offset logical drive creation was not blocked if any other logical drive in the array was not optimal.
 - *Exposure:* Create Logical Drive menu.
 - *Fix:* Block offset logical drive creation if any other logical drive in the array is not optimal.
- Fixed an issue where the drive firmware was being immediately activated for Mode E.
 - *Root cause:* Activation command was sent for Mode E.
 - *Exposure:* Drive FW Update menu.
 - *Fix:* Do not send activation command for Mode E firmware update.
- Added a numeric configuration option to set BMC I2C slave address.
 - *Root cause:* New menu to configure BMC I2C slave address.
 - *Exposure:* BMC Settings menu.
 - *Fix:* Numeric configuration option added to set BMC I2C slave address.
- Fixed an issue where Utility was reporting a successful drive firmware update on an invalid firmware image.
 - *Root cause:* SCSI status was not handled for firmware update calls.
 - *Exposure:* Drive Firmware Update menu.
 - *Fix:* Error-handling for firmware update will now consider SCSI status as well.
- Fixed an issue where the firmware version of the drive was not being updated after the drive firmware update operation.
 - *Root cause:* Latest inquiry data was not obtained from drive after the firmware update.
 - *Exposure:* Drive Firmware Update menu.
 - *Fix:* The physical drive will be re-queried for the latest inquiry data after firmware update completion.
- Fixed an issue where the firmware and build numbers were displayed inconsistently in tools.

- *Root cause:* The firmware version needs to be displayed as z.xy-w instead of z.xy bw.
- *Exposure:* Controller Information and Administration menu.
- *Fix:* The firmware version display was corrected to the format z.xy-w.

For the Legacy BIOS, this release provides the following fixes and enhancements:

- Fixed an issue where the device type of the backup array was set to "Split mirror orphan" after a split mirror operation.
 - *Root cause:* The configuration signature was hard-coded.
 - *Exposure:* Create Array menu.
 - *Fix:* The configuration signature was changed to a dynamic value based on the time returned from the system BIOS.
- Fixed an issue where CTRL-A allowed the user to set a drive as a spare drive that was under an erase operation.
 - *Root cause:* The drive under erase operation was not greyed out from drive selection list.
 - *Exposure:* Manage Arrays menu.
 - *Fix:* If the drive is under secure erase, the drive will be greyed out from the drive selection list.
- Fixed an issue where one physical drive can only be assigned to a RAID array as a dedicated spare drive in CTRL-A.
 - *Root cause:* Feature enhancement.
 - *Exposure:* Manage Arrays menu.
 - *Fix:* Multiple drives can now be assigned as a dedicated spare drive.
- Fixed an issue where the hot spare failed to assign to a RAID5 array created by MaxView.
 - *Root cause:* Incorrect Physical Drive(s) count was being sent to the firmware during Logical Drive creation.
 - *Exposure:* Manage Arrays menu.
 - *Fix:* Corrected the Physical Drive(s) count during the Logical Drive creation.
- Fixed an issue where CTRL-A showed an invalid array name after 26th Array when created through MaxView.
 - *Root cause:* Array names were not handled properly for displaying array names starting with AA (after 26th Array).
 - *Exposure:* In Manage Arrays and Disk Utilities menus.
 - *Fix:* Array names handled properly for names starting with AA onwards.
- Fixed an issue where ARCCONF was reporting "Logical Drives Not Contiguous".
 - *Root cause:* The transformation of logical drives was not initiated if a logical drive was deleted in the middle.
 - *Exposure:* Manage Arrays menu.
 - *Fix:* Initiated the transformation, if a logical drive in the middle is deleted.
- Fixed an issue where the firmware and build numbers were displayed inconsistently in tools.
 - *Root cause:* The firmware version needs to be displayed as z.xy-w instead of z.xy bw.
 - *Exposure:* POST and Controller Information menu.
 - *Fix:* The firmware version display was corrected to the format z.xy-w.
- Fixed an issue where unwanted variables and functions were not removed, causing memory corruption.
 - *Root cause:* Unwanted variables and functions were not removed.
 - *Exposure:* Set Boot Drive, Manage Arrays menu.
 - *Fix:* Code clean up. Fix issues include the following items:
 - Invalid entry "CN1:01:00 0KiB" is listed in Select Boot Device screen of CTRL-A utility.

- CTRL-A utility is stuck when creating a new logical drive.
- Garbage values in array details when new logical drive is created.
- CTRL-A Drives Selection displays blank screen or strange behavior after Secondary Boot device.
- Added support for BMC I2C slave address configuration.
 - *Root cause*: New feature.
 - *Exposure*: Controller Configuration menu.
 - *Fix*: A new menu BMC Settings is added to configure the I2C slave address.
- Fixed an issue with memory over run.
 - *Root cause*: Unwanted variables and functions were not removed.
 - *Exposure*: Set Boot Drive.
 - *Fix*: Code Clean up. Fix issues include the following items:
 - Invalid entry "CN1:01:00 0KiB" is listed in Select Boot Device screen of CTRL-A utility.
 - CTRL-A Drives Selection displays blank screen or strange behavior after Secondary Boot device.

2.2.2.9 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.2.1107/1.2.2.1104

For the UEFI/HII, this release provides the following fixes and enhancements:

- Fixed a failure seen with a SAS Drive firmware update from HII.
- Included the Microsoft signed UEFI driver.

For the Legacy BIOS, this release provides the following fixes and enhancements:

- Added support for selecting secondary OS Boot HBA Drive in "Select Boot Device" page of CTRL+A. Use <Ctrl+P> for selecting primary boot drive and <Ctrl+S> for selecting secondary boot drive.
- Fixed an issue where drives were not being correctly reported when configured as secondary boot drive.

2.2.3 Driver Fixes

2.2.3.1 Windows Driver Fixes

2.2.3.1.1 Fixes and Enhancements for Windows Driver Build 59002

There are no new fixes and enhancements for Windows drivers in this release.

2.2.3.1.2 Fixes and Enhancements for Windows Driver Build 58015

This release provides the following fixes and enhancements:

- Fixed an issue where device command timeouts occurred and caused a controller reset.
 - *Root cause*: When the host stopped sending sequential commands to a device, the driver command coalescing logic flag, which was based per controller, was not managed properly and caused a failure of sending coalesced commands to the device.
 - *Fix*: The coalescing logic flag is now based per device, instead of per controller.
 - *Risk*: Low

2.2.3.1.3 Fixes and Enhancements for Windows Driver Build 58012

There are no new fixes and enhancements for Windows drivers in this release.

2.2.3.1.4 Fixes and Enhancements for Windows Driver Build 57011

This release provides the following fixes and enhancements:

- Added support for Windows 2019 driver.

2.2.3.1.5 Fixes and Enhancements for Windows Driver Build 56008

This release provides the following fixes and enhancements:

- Fixed an issue where the copyright and provider information were incorrect.
 - *Root cause:* The copyright and provider were incorrect.
 - *Fix:* Updated the copyright and provider information.
 - *Risk:* Low

2.2.3.1.6 Fixes and Enhancements for Windows Driver Build 55022

This release provides the following fixes and enhancements:

- Fixed an issue where the IO would be lost during heavy IO, resulting in a controller reset.
 - *Root cause:* A race condition exists in MSI-X mode which would cause the command pointer to be cleaned up for the wrong handle.
 - *Fix:* Use the MSILock() for locking when using MSI-X.
 - *Risk:* Low

2.2.3.1.7 Fixes and Enhancements for Windows Driver Build 54013

This release includes the following fixes and enhancements:

- Fixed an issue where the Windows 2016 server would hang when the devices became unresponsive.
 - *Root cause:* Windows miniport driver was accessing the HW Soft Reset register irrespective of the controller type, resulting in a controller crash.
 - *Fix:* Added the check for HBA 1000 Series adapter and issue HW Soft Reset only to HBA 1000 Series adapter products.
 - *Risk:* Low

2.2.3.1.8 Fixes and Enhancements for Windows Driver Build 52013

This release provides the following fixes and enhancements:

- Command coalescing support for the RAID devices for small block sequential I/Os.
- PCIe slot number and host time are sent to the controller after it is reset by the driver.
- When I/O requests timed out, OS sends down task management requests such as LUN and target resets to the device. In such cases, the outstanding I/Os for that device are returned back to the OS as not completed to avoid further resets coming from the OS.
- Fixed an INF WHQL issue which is related to the new added registry items.
- Fixed an issue in the driver that caused the ARCCONF/maxView tools to wait forever for the commands sent to get the details of the adapter initiated events (AIF). For such commands, the driver should not consume the response, and this was causing application to wait forever. The response should be returned to ARCCONF/maxView instead.
- Fixed an issue in the driver that could result in the ARCCONF and maxView hang. The issue was a result of the SCSI Request Block (SRB) extension initialized incorrectly in the non-I/O path.
- Fixed an issue in the error recovery path that could lead to the OS crash. The issue was that the driver was holding on to a spinlock on a resource and causing a deadlock, while issuing the controller reset during controller recovery.

- Changed the company information from PMC to Microsemi.
- Fixed an issue where the driver incorrectly processed the returned value for processor number in the WHQL test "DP WLK– Hot Add-Device test – Verify driver support for Hot Add CPU test".
- Fixed an I/O timeout issue that was caused by a coalescing request that was not de-queued. Changed code to de-queue coalescing queue based on outstanding I/O of the specific logical drive.
- Fixed a Windows 2008 32-bit OS installation hang issue. There was a spin lock problem for 32-bit OS, which caused the system to hang. The fix was to change StartIo spinlock to Interrupt spinlock.
- Microsoft Windows recommends that for underrun case, SRB status should not be overrun. It should be set to SUCCESS (SRB stands for SCSI Request Block, a structure used to communicate the SCSI commands from the host to the controller).
- Fixed several issues related to issuing controller reset. These issues surfaced up as BSODs when the controller was coming out of the controller reset (the highest level reset a driver can do).
- During S3/S4 test, the firmware occasionally crashes due to the driver sending the wrong Init structure data to the firmware. During driver initialization, the driver gets its parameters from the registry by querying the kernel using the registry key. If there is no match, the code should terminate the search and return. If it is run beyond the buffer, and if the OS does not zero the buffer for the registry string, the driver may get a garbage value resulting in it sending a wrong value to the firmware, resulting in the crash.

2.2.3.1.9 Fixes and Enhancements for Windows Driver Builds 50666

This release provides the following fixes and enhancements:

- Fixed an issue where the system hung during Windows boot. Restricted the CPU node support to 8 nodes to ensure the data structure does not overflow if the system has a larger configuration.
- Updated product as "Adaptec SmartIO 2000" in the Windows INF file.

2.2.3.1.10 Fixes and Enhancements for Windows Driver Build 50663

This release provides the following Windows driver fixes and enhancements:

- Fixed an issue to report the correct number of scatter gather lists to the OS from the firmware.
- Added Windows system event logging for failures during the Windows driver initialization besides firmware panic conditions.
- Fixed an issue where references to the controller name displayed inconsistently across all the tools.
- Fixed an issue where lower IOPS performance was seen in HBA 1000 products compared with Series 8 products. This was partially caused by the driver hard coding the QD to 32 for the SAS targets. With this change, the driver sets the QD per target based on the controller firmware's settings.
- Added SAS Dual Port Device support. Fixed an issue with SAS device exposure issue.
- Added support for issuing soft resets.

2.2.3.2 Linux/VMware/XenServer Driver Fixes

2.2.3.2.1 Fixes and Enhancements for Linux, VMware, and XenServer Driver Build 59002

There are no new fixes and enhancements for Linux, VMware, and XenServer drivers in this release.

2.2.3.2.2 Fixes and Enhancements for Linux, VMware, and XenServer Driver Build 58012

This release provides the following fixes and enhancements:

- Fixed an issue where reading/writing data beyond LBA will trigger kernel panic.
 - *Root cause:* Inappropriate sense data being set would return the wrong value resulting in a kernel panic.
 - *Fix:* Proper sense data copied with the correct return values fixes the issue.

- *Risk: Low*
- Fixed an issue where the sysfs entries were not consistent between AACRAID and smartpqi drivers.
 - *Root cause:* Both drivers developed separately had only some fields in common and few additional fields which were not of primary interest.
 - *Fix:* Ensure consistency between the two drivers, and retain important values so that a script could be used to parse them.
 - *Risk: Low*

2.2.3.2.3 Fixes and Enhancements for Linux, VMware, and XenServer Build 57013

This release provides the following fixes and enhancements:

- Added retpoline support to correct potential spectre/meltdown issues for OEL 6.9 with both UEK4 and UEK3.
- Use command lsblk to show drives behind the controller are removable.
 - *Root cause:* The SCSI device removable bit was set to 1 in the driver code.
 - *Fix:* Added module parameter aac_removable (to pass while loading a module) which indicates whether the drive is removable or not. (aac_removable = 0: Not removable, aac_removable = 1: Removable)
- Fixed an issue where the build packaging changes and version numbers in modinfo and the rpm file did not match.
 - *Root cause:* To add a subversion (-1) after the version in UEK binaries. "-1" indicates a bump in driver version and the subsequent script changes will be "-2" and so on, when the driver code does not change.
 - *Fix:* Changed the Oracle Linux UEK spec file and the aacraid_build script to add an extra literal.
- Changed timeout in ARC to 60 seconds.
 - *Root cause:* IOP_RESET occurs more frequently.
 - *Fix:* Changed "AAC_ARC_TIMEOUT = 45" to "AAC_ARC_TIMEOUT = 60".
- Fixed an issue where the driver was sending IOP reset and causing the firmware to assert.
 - *Root cause:* The driver was sending IOP reset and causing the firmware to assert.
 - *Fix:* Issue drop I/O before changing interrupt mode to prevent firmware from asserting when sending IOP reset.
- Fixed an issue where using kickstart to install the OS leads to AACRAID driver install failure during pre-install steps.
 - *Root cause:* Pre-install would fail if the "/lib/modules/'uname -r'/updates" directory did not exist.
 - *Fix:* Removed the %pre section and the subsequent builds to be handled.
- Fixed an issue to prevent resets during Extended Error Handling scenario.
 - *Root cause:* Extended Error Handling mechanism gets triggered during error handling reset.
 - *Fix:* Check the bit that indicates if Extended Error Handling is enabled and required before any PCI BAR register access is done. If enabled, then disallow the regular error handling path.
- Added support for consistent naming convention for RPMs in UEK R3 and R4.
- Fixed an issue where the OEL/UEK matrix was broken.
 - *Root cause:* OEL/UEK matrix was broken.
 - *Fix:* Added UEK R4 support for OEL 7.2 and UEK R3 support for OEL 6.8.
- Fixed an issue where the controller could not detect the disk after doing a hot plug test.
 - *Root cause:* The OS was not exposing a drive when host_target_num=256.
 - *Fix:* Properly initialized a device element in the driver to allow a device rescan after hot-adding the drive

- Fixed an issue where ARCCONF stopped working after savesupportarchive was executed.
 - *Root cause:* All management commands are utilized at some point while executing the "arccconf savesupportarchive" command. So the driver was not able to perform the requested command.
 - *Fix:* Use aac_fib_free() api to free the management commands instead of aac_fib_free_tag() in aac_intr_normal().
- Fixed an issue where the Device Identification VPD page(0x83) was not supported for logical drive in SUSE11 SP3.
 - *Root cause:* VPD page(0x83) is not supported for SUSE kernel.
 - *Fix:* Added VPD page(0x83) for SUSE kernel.

2.2.3.2.4 Fixes and Enhancements for Linux Driver Build 56021

This release provides the following Linux fixes and enhancements (XenServer and VMware were not updated in this release):

- Fixed a hotplug issue with max id and rphy management.
 - *Root cause:* When a target device is removed from the topology, the driver did not clean up the sas_rphy (remote PHY) identifier in its mapping table. When a target device was subsequently added, the sas_rphy identifier can get reused resulting in two instances of it in the driver mapping table. When the OS sent an Inquiry command to the new target, the driver incorrectly used stale sas_rphy value and chose the incorrect target to send the command to. This resulted in the Inquiry command timing out and the target device was not exposed to the OS.
 - *Fix:* The driver will clean up the sas_rphy value in its mapping table when a device is removed so that stale data is not present in the table.
 - *Risk:* Low
- Fixed an issue where loading the out-of-box driver in Oracle 7.4 UEK4 was not being recognized.
 - *Root cause:* This was due to the incorrect override being specified in the depmod override.
 - *Fix:* Corrected the depmod override to use the correct UEK kernel.
 - *Risk:* Low
- Fixed an issue where both UEK3 and UEK4 could not be installed in the same environment.
 - *Root cause:* UEK3 and UEK4 could not be installed in the same environment.
 - *Fix:* The naming convention was changed in the spec files and the UEK variable was passed to the specfile and the iso Makefile.
 - *Risk:* Low
- The driver update disk did not contain the drivers for UEK kernels.
 - *Root cause:* The driver update disk did not include the UEK driver flavors after changing the naming convention.
 - *Fix:* Modified the procedure to include the UEK drivers for the driver update disk.
 - *Risk:* Low

2.2.3.2.5 Fixes and Enhancements for Linux and XenServer Build 56008 and VMware Driver Build 56009

This release provides the following fixes and enhancements:

- Fixed an issue where reset functions were broken up into their individual functions.
 - *Root cause:* The driver only had two reset functions, which were broken up into their individual functions for device, target, bus, and host.
 - *Fix:* This was an upstream patch, which was propagated into the out-of-box driver.
 - *Risk:* Low
- Fixed an issue where the smartctl could not get information from member drives of a RAID.

- *Root cause*: The channel was being passed in as 0, while for SmartIOC 2000 the channel should be 1.
- *Fix*: The fix is to change the channel from 0 to 1.
- *Risk*: Low
- Added a module parameter to hide the internal SEP device.
 - *Root cause*: The internal SEP can clutter the `/sys` file system.
 - *Fix*: Create a module parameter to be able to able to hide the internal SEP.
 - *Risk*: Low
- Fixed an issue where the reset function was not being sent in some situations.
 - *Root cause*: During reset processing, send Task Management function independent of which SCSI command.
 - *Fix*: Use the `reset_state` variable to track Task Management function.
 - *Risk*: Low
- Added support for `smp_utils` using `sas_transport`.
 - *Root cause*: The driver did not support `smp_utils`.
 - *Fix*: Included support for SMP functionality to the driver.
 - *Risk*: Medium
- Fixed an issue where the driver failed to add new devices on hotplug.
 - *Root cause*: Include error handling in hash add functions, and return error to caller. Include support to add the device at the beginning, so we do not add duplicates.
 - *Fix*: Reworked the hash add function and its callers to ensure that errors are detected early.
 - *Risk*: Low
- Added support for improved error handling for HBA commands.
 - *Root cause*: Include a flag in the controller command to indicate that this HBA command has an associated SCSI command.
 - *Fix*: Included flag to indicate that a SCSI command is associated with an HBA command.
 - *Risk*: Low
- Fixed an issue where out-of-bounds array references were possible for SmartIOC controllers.
 - *Root cause*: Referencing array elements prior to range checking could lead to undefined results.
 - *Fix*: Include array bounds checking prior to accessing arrays for targets.
 - *Risk*: Low
- Harmonized code paths for using `sas_transport`.
 - *Root cause*: The code paths for `sas_transport` and `non-sas_transport` were distinct.
 - *Fix*: Harmonized the code paths to be in one code path.
 - *Risk*: Low
- Backported inbox changes from 50740 to an out-of-box driver.
 - *Root cause*: In order to harmonize in-box/out-of-box drivers, sync changes between the two drivers.
 - *Fix*:
 - *Risk*: Low
- Fixed an issue where some devices may be offline after a reset.
 - *Root cause*: Devices which caused errors may be offline after reset.
 - *Fix*: Reset offline devices to be available after reset.
 - *Risk*: Low

- Increased timeout for SmartIOC controllers to 180 seconds.
 - *Root cause:* In some cases, timeouts have occurred for IO which would be successful.
 - *Fix:* Increased timeout to 180 seconds.
 - *Risk:* Low
- Reworked the IOP_Reset functionality.
 - *Root cause:* In-box driver 50834 included changes to the IOP_Reset functionality.
 - *Fix:* Backported these changes to the inbox driver.
 - *Risk:* Low
- Fixed an issue where the ARCCONF reset command could hang.
 - *Root cause:* Need to ensure that we are the only user of ioctl commands.
 - *Fix:* Protect access using the ioctl mutex.
 - *Risk:* Low
- Fixed an issue where the ARCCONF does not respond to commands after lockup on second controller.
 - *Root cause:* Check for controller failure during ioctl processing.
 - *Fix:* Included a check for controller health prior to executing ioctl.
 - *Risk:* Low
- Fixed a VMware issue that was causing the driver to fail.
 - *Root cause:* Some Linux functions are not available on VMware.
 - *Fix:* Implement compatibility functions for VMware.
 - *Risk:* Low
- Fixed an issue where a controller would unexpectedly reset due to a lack of resources.
 - *Root cause:* Some controller commands would not be returned to the management free pool.
 - *Fix:* Included a check to determine which pool a controller command belongs to and return appropriately.
 - *Risk:* Low
- Fixed an issue where kdump would not function on RHEL 7.1.
 - *Root cause:* Kdump on RHEL 7.1 would only allow 30 seconds prior to issuing a SIGKILL, which caused the driver to report an error.
 - *Fix:* Included a workaround for RHEL 7.1, which would not be interruptible during kdump.
 - *Risk:* Low
- Backported a timespec64 change from the Linux kernel.
 - *Root cause:* The current time methods would rollover, due to 32-bit values.
 - *Fix:* Use the 64-bit time methods.
 - *Risk:* Low
- Hardened support for AER errors.
 - *Root cause:* Include support to handle all soft/hard AER error.
 - *Fix:* Included additional support for both fatal/non-fatal AER errors, as well as enhanced error handling (EEH) errors.
 - *Risk:* Low

2.2.3.2.6 Fixes and Enhancements for Linux, XenServer, and VMware Driver Build 55022

This release provides the following fixes and enhancements:

- Fixed an issue where a driver could crash during heavy IO runs with 64 RAID volumes.

- *Root cause:* A call trace was seen when running heavy IO on 64 containers, which included driver re-entry after completing a command.
- *Fix:* Use atomic memory allocation for the buffer, as it is in interrupt mode.
- *Risk:* Low
- Fixed an issue where the controller was not reflecting the current time or the correct slot.
 - *Root cause:* When the `aac_send_bmic()` command was rewritten, the input parameter was treated as an output parameter.
 - *Fix:* Copy the input parameter to the buffer before sending the command.
 - *Risk:* Low
- Added support for drives larger than 2 TB
 - *Root cause:* SmartIOC 2000 did not recognize volumes larger than 2 TB. There was an invalid assumption that 2 TB support was tied to variable block size, which was not correct.
 - *Fix:* Use the correct attribute, new comm 64, to determine 64-bit support.
 - *Risk:* Low

2.2.3.2.7 Fixes and Enhancements for Linux, XenServer, and VMware Driver Build 54013

This release includes the following fixes and enhancements:

- Added support for XenServer 7.0.
- Added support for Ubuntu 16.04.2.
 - Kernel: 4.4.0-62-generic
- Added support for Ubuntu 14.04.5.
 - Kernel: 4.4.0-31-generic
- Added support for RHEL 7.3.
 - Kernel: 3.10.0-514
- Added support for Oracle Enterprise Linux 7.3.
 - Kernel: 3.10.0-514.el7.x86_64
 - UEK version: 4.1.12-61.1.18.el7uek.x86_64
- Added support for SLES SP2.
 - Kernel: 4.4.21.69-default
- Fixed an issue where the system crashed with the latest AACRAID driver 1.2.1-53005 for Linux kernel 3.10.101.
 - *Root cause:* There were two issues: one, the devices were all not freed when the driver was removed; two, all the devices were not being freed since expanders were allocated as `sas_expander` and then changed to a different class called end device. Now when the driver is removed, the kernel skips over the `sas_expander` class and the kernel complains.
 - *Fix:* Ensured that SAS transport is released and that all expanders are assigned to the end-device type only.
 - *Exposure:* HBA 1000 Series adapter products running Linux kernels 3.10.x (such as Debian 8.1, RHEL 7.3, and OL 6.8)
 - *Risk:* Medium
- Fixed an issue where ISSCSI shows drives information, which should be part of Array.
 - *Root cause:* After creating RAID in SmartIOC Mixed Mode, the drives are still shown in ISSCSI output as `/dev/sdx`. After Creating RAID in SmartIOC Mixed Mode, the drives that are part of the RAID are not removed. This is because `sas_remove_device` is not called.

- *Fix:* After creating the RAID in mixed mode drive expose flag value is changed by the firmware. Based on the expose flag value a check is added in the `aac_scan_host` to call the `sas_remove_device` function.
- *Exposure:* SmartIOC2000 products configured in Mixed Mode of operation.
- *Risk:* Low
- Added improvements to the controller reset recovery.
 - *Root cause:* Several areas were improved to make the controller reset recovery procedure more reliable and responsive.
 - *Fixes:*
 1. Reduced the time to initiate the recovery process by polling the controller health more frequently (1 minute).
 2. Streamlining of the reset handler to avoid multiple controller resets sent from the driver in the event of controller hang.
 3. The driver logs the controller reset initiation into `/var/log/messages`.
 4. Reduce the burden on the recovering controller by not sending any sync commands when the controller is in a faulty or recovering state.
 5. Reduced the time-out duration for the sync timeouts to enable the recovery handlers to kick in in a timely manner.
 - *Exposure:* HBA 1000 Series adapter on VMware and Linux OSes.
 - *Risk:* Medium
- Added support to improve the packaging of the DKMS driver packages for the RPM installer.
 - *Root cause:* The older build scripts required manual intervention to build the DKMS packages for RPM.
 - *Fix:* Created the DKMS RPM-based driver installer, which does not require any additional user action to build or install.
 - *Exposure:* HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where the Linux build script was not building only for the current kernel.
 - *Root cause:* In the Ubuntu driver build script, there is no check to make sure it builds the driver only for the installed kernel version; this causes the build script to build drivers for all the kernel modules and headers installed.
 - *Fix:* A check was added in the Ubuntu driver build path to check if proper kernel headers and modules are installed and to build the driver only for the kernel version installed and running.
 - *Exposure:* Applies to Ubuntu Linux drivers for HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where the HBA 1000 16i sysfs output was in multiple lines.
 - *Root cause:* The sysfs flags entry for the driver display the compile and configuration flags as strings in multiple lines.
 - *Fix:* Each of the sysfs configuration parameters were assigned in a 64-bit bitmap that could be fitted into a single line.
 - *Exposure:* Applies to all Linux drivers for HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where the kernel prevents the driver from being removed.
 - *Root cause:* The reference count of the device exposed by the driver is not zeroed.
 - *Fix:* The driver ensures that all of the allocated memory is freed properly and the reference count is zeroed.

- *Exposure:* HBA 1000 Series adapters.
- *Risk:* Low
- Fixed an issue where the 4K sequential write performance of the Seagate 6 TB SATA HDD was lower than expected.
 - *Root cause:* The native 4K sector drive was reported as not being an IOBypass drive, which made the driver set the maximum device queue depth to 256.
 - *Fix:* The QD is now set based on the device type, as reported by the device itself.
 - *Exposure:* Linux drivers for HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where XenServer 6.5 crashed while running IOs after extended periods of testing.
 - *Root cause:* There was a memory allocation failure from XenServer when using ISA memory DMA specifier.
 - *Fix:* Removed ISA memory DMA specifier, and replaced with GFP_DMA32 where applicable.
 - *Exposure:* XenServer 6.5 driver for HBA 1000 Series adapters.
 - *Risk:* Low
- Provided a fix to terminate the driver thread when it is stuck on a periodic sync command (such as time sync that is sent to the firmware), resulting in a system hang.
 - *Root cause:* When the driver thread performs a periodic time sync and if the firmware is crashed, during that time the driver thread waits for the response that would never arrive from the firmware. This also blocks the controller recovery from happening because it blocks the reset path from the kernel.
 - *Fix:* The fix is to allow SIGTERM to terminate the driver thread, resulting in the error handlers to continue the controller recovery.
 - *Exposure:* All Linux drivers for HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where the drive was being marked offline when an IOP reset was performed in the middle of a heavy I/O.
 - *Root cause:* When the SCSI mid layer marks a drive offline due to repeated failures, there was no handling in the driver to clear the offline state and/or let the drive be configured again.
 - *Fix:* The fix is to identify any drive that has been marked offline by the SCSI layer and remove the device. Then invoke a bus rescan so that the device and any further changes in the configuration can be detected and get updated in the Linux driver device map (as well as the layers above).
 - *Exposure:* All Linux drivers for HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where I/O timeouts with a value of zero could potentially be created, causing premature timeout of the I/O.
 - *Root cause:* Previously the time out value was set by default to 1 second, which can be too short in cases where devices are misbehaving.
 - *Fix:* If a zero timeout is detected, set the timeout to 30 to allow for reasonable time to complete.
 - *Exposure:* VMware drivers for HBA 1000 Series adapters.
 - *Risk:* Low
- Fixed an issue where there was a potential zero memory access if the IOP reset did not go through properly.
 - *Root cause:* On IOP RESET, the PCI Memory and IO space are freed up. If the controller is not operational after the controller reset, there is a potential issue for null pointer access in the periodic time sync loop.
 - *Fix:* Terminate the periodic loop if the controller is not restored to a healthy state.
 - *Exposure:* All Linux drivers for HBA 1000 Series adapters.

- *Risk: Low*

2.2.3.2.8 Fixes and Enhancements for Linux and XenServer Build 53005 and VMware Driver Build 53004

This release provides the following fixes and enhancements:

- Fixed an issue where the OS kernel panicked when removing the AACRAID driver.
 - *Root cause:* The driver sends wellness commands periodically to the adapter firmware. The commands also contain the driver version as well. When a custom driver is used (as in this case), the driver version size is incremented to beyond what is supported by the command ("custom" string is added to the version). This causes a buffer overflow, and, when the driver is removed, it manifests in OS kernel panic.
 - *Fix:* Ensure that the driver version size is below the permitted size (use "src" instead of "custom").
 - *Exposure:* Customers who compile their own versions of the AACRAID drivers and use it with HBA 1000 Series adapters.
 - *Risk: Low*
- Fixed an issue where the driver was reporting the adapter dead in case of a SCSI hang.
 - *Root cause:* When the driver receives an "eh abort" request (SCSI hang) from the kernel, it checks for the status of the adapter. If the adapter's `SELF_TEST_FAILED` condition is set, the driver returns without resetting the controller. This causes the kernel to offline the adapter.
 - *Fix:* First check for adapter `KERNEL_PANIC` before checking the other conditions. This works because if a `KERNEL_PANIC` is detected, the driver resets the adapter.
 - *Exposure:* All customers.
 - *Risk: Low*
- Fixed an issue where controller firmware lockup was encountered on the system while deleting units.
 - *Root cause:* When the driver does not explicitly clear the `AdapterProcessed` flag, it causes the firmware to lockup.
 - *Fix:* Ensure `AdapterProcessed` is not set when submitting commands.
 - *Exposure:* All customers using HBA 1000 Series adapters.
 - *Risk: Low*
- Fixed an issue where customers on Ubuntu/Debian distributions did not have the documentation for building and installing the driver from the source.
 - *Root cause:* The ARC driver source is distributed in src rpm format to customers. The instructions for compiling src rpm on non-rpm distributions were not provided and non-rpm distributions did not have the required environment setup to use the src rpm files.
 - *Fix:* The driver source is now also distributed in tar gz format along with instructions to install the driver from source.
 - *Exposure:* Customers who compile and install the driver (as opposed to those using binary release drivers).
 - *Risk: Low*
- Added support to include instructions detailing the installation process of DKMS deb driver packages.
 - *Root cause:* The ARC driver release did not contain documentation on the installation methodology of DKMS deb packages.
 - *Fix:* Added DKMS deb driver installation documentation.
 - *Exposure:* Customers using DKMS deb driver packages.
 - *Risk: Low*
- Fixed an issue where VMware was displaying a PSOD (purple screen) when the driver triggered an IOP reset.

- *Root cause:* When the driver issued an IOP reset, it released all the memory and additional resources it had and then reinitialized the driver once the reset went through. When it released the memory, it freed up fib DMA memory at an incorrect misaligned memory location.
- *Fix:* Freed the memory at the correct DMA memory location. Fixed an additional memory leak issue.
- *Exposure:* Customers using VMware.
- *Risk:* Low
- Added sysfs entry to expose the driver version.
 - *Root cause:* There was no easy way to retrieve the currently running driver version.
 - *Fix:* Added sysfs entry to expose the driver version.
 - *Exposure:* Customers who need the currently running driver version without resorting to management tools.
 - *Risk:* Low
- Fixed an issue where the driver was not recovering a hung controller.
 - *Root cause:* Driver would get caught in an endless loop resetting the card.
 - *Fix:* Cleared the Health/Status register before resetting the card. Added state to transition to `KERNEL_BOOTING` when the firmware starts. Allowed to stay in state until `KERNEL_UP_AND_RUNNING` is set.
 - *Exposure:* All customers.
 - *Risk:* Low
- Removed PCI ID of controllers that are not supported by this driver.
 - *Root cause:* The current driver claimed support for all the future generation of products, even though those products have their own driver.
 - *Fix:* Removed PCI ID of future generation cards.
 - *Risk:* Low
- Fixed driver SCSI hang on "eh reset" request from kernel.
 - *Root cause:* The driver has two pools of fibs (data transporting abstractions): one for kernel IO and requests, and the other for management commands. The issue was caused when the driver sent kernel IO-tagged fib to the management queue during error recovery.
 - *Fix:* Added pool tags for fibs to identify and prevent fibs from either pool from being sent to the wrong pool.
 - *Exposure:* All customers.
 - *Risk:* Low
- Added support for Oracle Linux (OL) 6.8.
 - *Root cause:* The driver did not support Oracle Linux 6.8.
 - *Fix:* Added support for Oracle Linux 6.8.
 - *Exposure:* Customers who require support for Oracle Linux 6.8.
 - *Risk:* Low
- Fixed an issue where kdump fails on RHEL 6.5 when it was installed on Microsemi adapter cards.
 - *Root cause:* Kdump is a Linux mechanism that allows for the capture of kernel logs and cores in case of kernel or driver crash. On centos/RHEL systems, kdump is loaded with only a single CPU enabled. When that happens, the Microsemi driver goes into a rudimentary interrupt (intx) mode (which would work in normal cases), but, due to an Intel CPU bug, interrupts are disabled/do not work in kdump mode. This bug prevents the driver from loading and the kernel logs and core from being saved.
 - *Fix:* To solve this issue, the driver forces full interrupt mode when it detects the driver is in kdump context and the number of CPUs is 1. This enabled the driver to fully load.
 - *Exposure:* Customers who use RHEL/Centos and kdump.

- *Risk*: Low
- Fixed an issue where customers were not able to install the driver on rpm-based distributions (RHEL/OL/etc.) with custom kernels.
 - *Root cause*: Currently the ARC driver supports various Linux kernel versions by creating binary drivers for any particular kernel. When a customer uses a custom kernel and if we do not have access to that custom kernel and its headers, it would be difficult to create a driver that supports that kernel. In addition, the customer depends on us for support whenever they upgrade the kernel. The alternative is to use a DKMS driver that would support all the custom kernels. Deb DKMS drivers are already supported; need to add support for rpm DKMS drivers.
 - *Fix*: Added support to generate rpm DKMS drivers.
 - *Exposure*: Customers who use custom kernels on rpm-based OS distributions.
 - *Risk*: Low
- Added support for Oracle Linux UEK3 and UEK4 kernels.
 - *Root cause*: Currently, the driver supports UEK3 and UEK4 kernels indirectly by supporting the Oracle Linux (OL) distributions that contain either UEK3 or UEK4 kernels. So when a customer needs a particular UEK version driver, they have to use an OL distribution driver version that supports the required UEK kernel version.
 - *Fix*: Added support to generate OL UEK3 and UEK4 specific binary drivers.
 - *Exposure*: Customers who require Oracle Linux UEK3 or UEK4 specific drivers.
 - *Risk*: Low
- Fixed potential null pointer exception in case of IOP reset failure.
 - *Root cause*: On IOP RESET, the PCI memory and IO space are freed up, and once the controller is detected to be up and operational, the initialization path is invoked to recreate the IO path resources. If the controller is not operational, the IOP reset path returns with a failure. The command thread invokes check health that relies on the IOP reset mechanism to recover the controller. If the call path returns with an error, however, the driver interprets it incorrectly and continues. It then erroneously pokes a resource pool which may have been freed up by the incomplete reset mechanism in the previous step.
 - *Fix*: Make sure the driver does not poke the freed resource pool in case of an improper IOP reset.
 - *Exposure*: All customers.
 - *Risk*: Low
- Added support for Oracle Linux (OL) 7.1.
 - *Root cause*: Driver did not support Oracle Linux 7.1.
 - *Fix*: Added support for Oracle Linux (OL) 7.1.
 - *Exposure*: Customers who require support for Oracle Linux 7.1.
 - *Risk*: Low
- Added support to ensure proper timeout when the timeout set to 0.
 - *Root cause*: The kernel at times sends IO commands which at times have a timeout value of 0, which is very short for any drives (HDD or otherwise).
 - *Fix*: The driver uses a timeout of 30 seconds if it detects timeout as being set to 0.
 - *Exposure*: All customers.
 - *Risk*: Low
- Fixed an issue where the driver `eh_reset` handler times out while waiting for the driver thread to stop.
 - *Root cause*: When the adapter crashes or does not respond, the driver tries to send a reset command. In parallel the driver kthread keeps sending periodic sync commands, and because the adapter is not in the correct state, it gets stuck in a fault state. This internally happens when the driver tries to stop its kthread from running but is hung waiting from a response from the adapter.

- *Fix:* Allow `eh_reset` to send a kill signal to the kthread, and then allow `eh_reset` to start a controller reset.
- *Exposure:* All customers.
- *Risk:* Low
- Fixed an issue where drives were being marked offline when IOP reset was performed in the middle of heavy IO.
 - *Root cause:* The Linux kernel marked drives offline when the adapter crashed in the midst of a heavy IO. These drives did not come back even after the adapter was reset.
 - *Fix:* Identify any drives that have been marked offline by the kernel and then remove the device. Then invoke a bus rescan in order for the device to show up in the system.
 - *Exposure:* All customers.
 - *Risk:* Low
- Improved the controller reset/recovery mechanism in Linux.
 - *Root cause:* The driver took 30 minutes to identify whether the adapter has crashed (which is not desirable). The reset logic does not care if the adapter is healthy before issuing an IOP reset, which would cause the request to timeout.
 - *Fix:* Reduced the health check interval from 30 minutes to one minute. The reset logic includes the health check code in determining the IOP reset. It is necessary to avoid a soft reset mechanism and trigger a hardware reset after giving a one minute delay to allow firmware log capture.
 - *Exposure:* All customers.
 - *Risk:* Low
- Added support for SAS transport in the driver.
 - *Root cause:* Customers requested support of SAS transport.
 - *Fix:* Added rudimentary support for SAS transport according to customer requirement. Currently has support to expose the SAS address of the drives and expanders, and to enable them as SAS devices.
 - *Exposure:* Customers who need SAS transport functionality.
 - *Risk:* Low

2.2.3.2.9 Fixes and Enhancements for Linux, VMware, and XenServer Driver Build 52011

This release provides the following fixes and enhancements:

- Added script to upload DKMS package to Ubuntu PPA.
- Added support for Ubuntu 16.04.
- Added support for RHEL6.8.
- Added support to build DKMS deb packages.
- Add support for Ubuntu 14.04.4
- Add support for SLES 12 SP1
- Fixed an issue that caused the IRQ warnings during the driver initialization. The warning message during IRQ request was due to the usage of a deprecated flag, `IRQF_DISABLED`.
- Fixed an issue where the driver was hanging when the kernel crash and `kdump` were invoked from `sysfs`. The root cause was that the controller was in MSI-x mode, while the crash kernel driver was expecting a response from the firmware in INT-x mode.
- Under Oracle Enterprise Linux 6.5, when the kernel got updated to 3.8.13-68.1.3.el6uek.x86_64, the `initramfs` was not re-built correctly, which resulted in a crash. The issue was that the path to the driver file in `depmod.d` was wrong and has been fixed.
- Fixed an issue after a controller reset where the host time and driver versions were not reset during the controller-driver handshake.

- Implemented Soft Reset to recover controller in case IOP RESET command fails. Soft Reset is the highest level reset that a driver can do.
- Fixed an issue where the firmware crashed after the driver sent a reset request.
- Fixed an issue where the PCIe slot number information was not displayed by the tools after the controller reset. This was because the PCIe slot number was not sent to the firmware after the controller reset, and the driver now saves the slot number before reset and re-issues it to the firmware after the reset is complete.
- Fixed an issue where the driver crashes on loading into kernels above 3.18 and below 3.19. When SCSI tag support was added, the function `scsi_activate_tag`, which is required to enable the SCSI tags, did not enable it for kernels above 3.18 and below 3.19. If a kernel that is in between these two versions is installed, the driver crashes. Changed the macros to cover up to, but not including, 3.19.
- ARCCONF fails in getting controller information for HBA controllers in large configuration, (e.g., more than 90 drives). The fix was to correct the SCSI channel ID assignment.
- Fixed an issue where there was an OS kernel panic when the controller driver was removed. The wellness command buffer was overrun due to larger driver version string that was inserted when the driver was built from the source rpm package. The fix was to make the protective copy of the driver version within size limits and change the default source driver version string to "src".
- Reordered error checks in `aac_src_check_health`, so that `SELF_TEST_FAILED` is checked after `KERNEL_PANIC`. This was preventing the controller resets from being sent down to the controller in the event where the controller reset was the only option to recover the controller.
- Added support to build script to create tgz src packages.
- Updated readme.txt correct steps required for DKMS installation.

2.2.3.2.10 Fixes and Enhancements for Linux, VMware, and XenServer Driver Build 50663

This release provides the following Linux, VMware, and XenServer driver fixes and enhancements:

- Fixed the issue related to the co-existence of Series7/8 and Smart-family controllers.
- Added macros to log driver and function names to default print functions.
- Added Linux system (dmesg) logging for driver initialization failures.
- Fixed an issue in the driver logic to ensure expander device that were hot-added or removed are reported to the OS.
- Added support for Citrix XenServer 6.5.
- Fixed an issue when a PCI error was detected, an IOCTL command was sent to the firmware from the management software, causing undefined behavior on the management software side.
- Set queue depth value to the firmware reported drive queue depth value.
- Fixed an issue where SuSE 11 SP3 failed to detect the correct boot partition.
- Added support for RHEL 7.2, SLES 12 SP1.
- Fixed an issue where the RHEL 6.6 installer could only see part of the logical drive when the LD size was bigger than 2 TB.
- Fixed an issue with Ubuntu 14.04.3 reporting the wrong LD size on a device greater than 2 TB.
- Added support to send the correct maximum MSI-X count during EEH recovery.
- Added a check before freeing the PCI-consistent memory.

2.2.4 Management Software Fixes

2.2.4.1 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 3.02.00 Build 23699

There are no new fixes and enhancements for maxView Storage Manager/ARCCONF in this release.

2.2.4.2 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 3.02.00 Build 23600

This release provides the following fixes and enhancements:

- Fixed an issue where ARCCONF was unresponsive after device firmware update with 256K chunk size.

- *Root cause:* Providing unsupported chunk size as input to IMAGEUPDATE command resulted in unresponsive controller.
- *Fix:* Added changes to validate the input chunk size and provide a valid response when user provides with invalid chunk size.
- *Risk:* Low
- Fixed an issue where SMART data is not displayed for physical devices in standalone mode of maxView.
 - *Root cause:* The response buffer was corrupt due to invalid memory access resulting in failure to display the information in maxView with Standalone mode.
 - *Fix:* The invalid memory access is fixed to send the proper response buffer to be displayed in maxView for SMART data.
 - *Risk:* Low

2.2.4.3 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 3.00.00 Build 23488

This release provides the following fixes and enhancements:

- Fixed an issue where a warning message was missing when a customer created multiple logical devices on the same physical devices.
 - *Root cause:* Need to add a warning if multiple logical devices are created on a device that is online.
 - *Fix:* Added a warning if multiple logical devices are created on a device that is online.
- Fixed an issue where the default size was not accepted for the expansion of RAID1 created with SSDs in maxView.
 - *Root cause:* While expanding a logical device from maxView, max size and user input size were not compared properly and the default value was not accepted.
 - *Fix:* Made the changes to compare the max size with the user input size to accept default size.
- Fixed an issue where RAID level migration by specifying the size as MAX was failing in ARCCONF.
 - *Root cause:* Validation of stripe size, size, RAID level, and legs parameters were missing in ARCCONF while blocking the modify command.
 - *Fix:* Added validation based on stripe size, size, RAID level, and legs parameters for RAID level migration.
- Fixed an issue where lane information was missing in maxView.
 - *Root cause:* New implementation to display the lane information of connector maxView.
 - *Fix:* Added changes to display the lane information Connector tab in maxView.
- Fixed an issue where creating a logical device failed in German Windows server.
 - *Root cause:* In maxView, an exception while formatting the size in German locale failed to create a logical device.
 - *Fix:* Made changes to format the size according to locale while populating the size tool-tip in maxView.
- Fixed an issue where maxView email notification was not working.
 - *Root cause:* Instead of TLS, the mail server was using deprecated SSL, which failed to deliver the email notifications.
 - *Fix:* Added changes to use SSL as a fallback mechanism for secure communication for the mail server when TLS connection fails.
- Fixed an issue where firmware upgrade for Micron 5100 SATA device failed.
 - *Root cause:* Mode 3 support is failing for SATA device firmware upgrade due to wrong parameters for ATA passthrough command.
 - *Fix:* Made changes to provide valid input parameter for modes 3 and 7 for SATA firmware upgrade through ATA passthrough command.

2.2.4.4 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.06.00 Build 23164

This release provides the following fixes and enhancements:

- Fixed an issue where an ARCCONF segmentation fault was observed during array creation using SSD in Ubuntu 16.4.3.
- Fixed an issue with the setboot device in the Web Help menu.
- Fixed an ARCCONF general failure while performing a modify array operation.
- Fixed an issue where there was a difference in logpath for maxView/ARCCONF savesupportarchive OS log collection.
- Fixed an issue where the maxView controller firmware update summary page showed as unknown instead of upgrade.

2.2.4.5 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.05.00 Build 22932

This release provides the following fixes and enhancements:

- Added support for executing UART commands through ARCCONF.
- Added a new command in ARCCONF to enable/disable SMART poll on the controller for RAW drives.
- Fixed an issue where exceeding the backup unit temperature caused a permanent warning sign to display on the controller.
 - *Root cause:* There was no check to not render the warning icon when the CAP temperature in the backup unit went back to normal.
 - *Fix:* When the CAP temperature is back to normal, the rendering of the orange warning icon for CAP temperature is blocked.
 - *Risk:* Low
- Fixed an issue where every user was being assigned an administrator role regardless of their original role.
 - *Root cause:* Using the domain name, every user was getting the Admin role irrespective of his original role.
 - *Fix:* Correct user roles are identified using domain-level authentication as well.
 - *Risk:* Low
- Added support for Sanitize Freeze and Sanitize Anti-Freeze lock.
- Added SSD IO Bypass option for arrays created with SSDs.
- Added description for T:L in ARCCONF GETCONFIG command.

2.2.4.6 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.04 Build 22665

This release provides the following fixes and enhancements:

- Added support for vSphere 6.5 web client plugin.
- Added support for XenServer 7.0.
- Fixed an issue where maxView was not displaying Tape drive model and negotiated transfer speed.
 - *Root cause:* ATTR_NAME_DRIVE_MODEL was not available for the tape drive due to an unknown model name.
 - *Resolution:* Added mapping of product ID to the model so that the model name is shown properly in maxView and ARCCONF.
 - *Risk:* Medium
- Fixed an issue where maxView was not displaying the firmware version during controller firmware upgrade.
 - *Root cause:* New firmware version and build info file pointers was changed by the firmware.

- *Resolution:* Added the new index and changed the logic for backward compatibility as well.
- *Risk:* Low
- Fixed an issue where the logical device cannot be deleted by ARCCONF when the logical device is in a reconfiguring state.
 - *Root cause:* Delete was blocked if the logical device state was migrating.
 - *Resolution:* Added warning for reconfiguring and unblocking delete call for Smart controllers.
 - *Risk:* Low
- Added the feature bit-based stripe size option display.
 - *Root cause:* 8K stripe size was not supported by firmware.
 - *Resolution:* Added feature bit-based stripe size option display.
 - *Risk:* Medium

2.2.4.7 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.03 Build 22476

This release provides the following fixes and enhancements:

- Enabled the configuration of the idle session timeout in maxView.
- Added support for Mode E and Mode F, along with help for expander upgrade in maxView and ARCCONF.
- Added method name, arguments, and the response data in a format for calls made to CIM in the CIMAdapter.log. Removed unwanted log messages.
- Added support for the slot information to be displayed for a drive that has a valid secondary location(slot) irrespective of enclosure for Smart Controller.
- Added code for the following items:
 - Dumping logs in Custompath.
 - Get GOS GUI and agent logs.
 - Get particular folder logs.
- Added proper message text when attempting to create RAID 10 when there is not enough space on an array.
- Added the proper check for verifying rebuild task and assigning the task string.
- Removed additional space from the set properties string in popup message box.
- Upgraded to latest version of JRE to address security issues.
- Modified check to verify if connector functional mode and pending functional mode are not in HBA mode.
- Modified code to allow 128 drives based on ADU report where it is maintained as 128.
- Added changes in storlib and ARCCONF to support Mode 3 support of SATA drive.
- Added a new Boolean "temperatureEventsGenerated", which will be set to true on the first abnormal temperature event and will be set to false once the temperature reaches normal again. This will allow one event each to be generated whenever the temperature goes from normal to abnormal and vice versa.
- Made the following name changes:
 - Changed "Hard drive access name" to "Disk Name" to sync with the logical drive and physical drive.
 - Added proper validations to display "Disk Name".
- Added a WWID of the SEP device for enclosure of the logical ID instead of from the expander device.
- Added code to capture all debug logs from controller.
- Fixed the following UX issues in the Enclosure log:
 - Fixed Enclosure power supply
 - Fixed Enclosure ID issue

- Added smart controller check for `task` and `uninit` commands.
- In ARCCONF, fixed an issue where `getConfig 1 ld` command shows data type and interface type as "Not Applicable"; changed to display "Device Type".
- Fixed an issue where maxView did not detect the controller; fixed the serial number and SES descriptors to fall under ASCII character set.
- Fixed an issue in maxView where the controller could not be found after installing a configured drive when the server was powered off.

2.2.4.8 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.02 Build 22404

This release provides the following fixes and enhancements:

- Implemented code for `REQUERY_PHYSICAL_DEVICE` for expander, and physical drive firmware upgrade.
- Added support for specifying the PMC BMC Service Interface (PBSI) I2C address through maxView and ARCCONF.
- Added support for "save" and "playconfig" options for HBA Products, allowing users to preserve the configuration when they replace the controller.
- Added support for maxView/ARCCONF to display the new firmware image version for the 82885T expander by reading the firmware image metadata. This enables the customer to know the version of the firmware image which they are flashing.
- Fixed an issue where maxView in Linux was reporting a failure to retrieve local IP address.
- Added support in ARCCONF to provide more outputs on unused and reserved size of physical drives.
- Added connector information to `controller_config.txt` support archive file.
- Refactored progress percentage update in maxView GUI, when multiple progress tasks are running in the controller.
- Added support to generate a firmware upgrade completed event as a client-generate event when the firmware upgrade is performed through the GUI.
- Added support to disable the create logical ribbon icon when no ready drive or optimal drive with enough free chunk is available.
- Added proper error message when renaming a logical drive which is part of the backup array.
- Added support to move the Save and Restore configuration operation from the controller node to the system node.
- Fixed an issue where the support archive could not be saved in the specified directory with ARCCONF (Linux).
- Fixed an issue where the expander firmware in maxView failed to upgrade.
- Fixed an issue in maxView where the "Change drive type" operation failed with the following error message: "No physical drive with required size and/or interface type"
- Fixed an issue where there were the wrong limits of "Expand a LD to a new array" and "Move array to a set of new PDs" in maxView.
- Fixed an issue where maxView incorrectly prompted a successful "split mirror backup" operation on the 64th logical drive.
- Fixed an issue where maxView build 22270 failed to delete logical drives in certain cases.
- Fixed an issue where "arccconf identify" failed with the following error message: "Invalid arguments"
- Changed the ARCCONF `expanderupdate` command default chunk size from 64k to 4k.
- Provided a fix to avoid listing the local system in the delete remote system dialog.
- Fixed an issue where the maxView shortcut icon was not available in the non-English (i.e., German, Spanish, Russian, Japanese, Chinese) version of CentOS 7.2.
- Added support for the firewall script to be changed for other languages in CentOS 7.2 (i.e., Russian, German, Spanish, Japanese, Chinese).
- Fixed an issue where the Adaptec Event Monitor did not start on reboot once enabled.

- Fixed an issue where the Adaptec Event Monitor Environment was not set up properly on Ubuntu.
- Fixed an issue where there were repeated events after setting boot type property.
- Fixed an issue where both maxView and ARCCONF were not working with XenServer 6.5.
- Fixed an issue where the ARCCONF `SMP_Phyerrorlog` returned the same information for all PHYs on expanders.
- Fixed issues in remote system adding.
- Fixed an issue where logical drive creation on an existing array failed when the logical drive size was given in MB from maxView.
- Fixed an issue where maxView failed to login after failing a drive through ARCCONF `setstate` command on Linux x86.
- Fixed an issue where the Hypervisor Guest OS maxView was not showing a progress bar under the task tab for any process.
- Fixed an issue where the ARCCONF "Move Array" operation failed .
- Fixed the syntax error message during maxView installation in CentOS 6.7
- Fixed an issue where the "Logical Drive Deleted Successful" message was displayed for partitioned logical drives in maxView, but logical device was not deleted
- Fixed an issue where there was a difference between ARCCONF `getConfig` output for a failed drive and an available drive.
- Fixed an issue where there were Linux issues in maxView when logging in as a standard user who has maxView admin access.
- Fixed an issue where a blank status dialog was shown when a logical device name was changed to more than 16 characters.
- Fixed an issue where ARCCONF gave a successful message on an invalid SEEPROM image.
- Fixed an issue where maxView showed an incorrect message after setting an invalid value for parallel consistency check count value.
- Fixed an issue where the locate array icon was disabled in maxView during an expand array operation.
- Fixed an issue where the logical creation summary page showed an array's name as unknown.
- Fixed an issue where the identify command with time option was not working.
- Fixed an issue where no sense data was reported when issuing ARCCONF Passthrough `read/writewithsense`.
- Fixed an issue in maxView where there was an unknown reason for a warning sign on the system level.
- Fixed an issue where the speaker status in maxView did not have enough space to show the field.
- Fixed an issue where the maxView login failed with invalid error message after clear configuration.
- Fixed an issue where maxView was not generating shortcut icons in SLES 11 SP3 OS and in the desktop in CentOS 5.10.
- Fixed an issue where maxView was not showing the progress bar for Hypervisor Guest OS.
- Added a proper error message for Locate Array.
- Fixed an issue to resolve login failure with error message "Controller firmware or Driver not responding" when there are no controllers connected.
- Fixed an issue where maxView displayed Rebuild Priority as "MediumHigh" when it is set to "Low".
- Fixed an issue where maxView displayed Expand Priority as "MediumHigh" when it is set to "Low".
- Fixed an issue where the "NRMFSADebug.log" was filled with many repeating entries in the Adaptec Event Monitor.
- Fixed an issue where the HDD firmware update failed from maxView/ARCCONF.
- Fixed an issue where a completed task was shown as running in maxView.
- Fixed an issue where the user was unable to delete the task log maxView.
- Fixed an issue in ARCCONF where `savesupportarchive` returned a "This PHY is vacant" message.

- Fixed an ARCCONF issue where `setconnectormode` failed when changing from RAID to Mixed Mode.
- Fixed the event and config update issue in maxView, when user changes multiple controller properties all at one time.
- Added a proper error message when moving logical drives from Large Array size to Small Array size.
- Fixed an issue where the Process Running symbol was not displaying in maxView for any operation.
- Fixed an issue where maxView showed i2c address as Not Applicable.
- Fixed an issue where the connector mode Changes Popup was not showing the expected message.
- Added a proper message for when HDDs and SSDs are used for logical drive creation in ARCCONF.
- Fixed an issue where the Navigate Tree icon was not displaying for spare drives in maxView array resources.
- Fixed an issue where there was a difference in the controller config log information between maxView support zip and ARCCONF support zip.
- Fixed an issue where maxView showed an incorrect message after setting an invalid value for parallel consistency check count value.
- Added a proper error message when renaming the logical drive that was part of the backup array.
- Fixed an issue to make ARCCONF/maxView maximum supported drive in an logical drive the same as in UEFI/SSA.
- Fixed an issue where password encryption failed for the maxView agent and web-based client plugin for VMware ESXi 6 on Windows machines.
- Fixed an issue where the i2c address was unable to be changed from ARCCONF on HBA1000 controller.

2.2.4.9 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.01.00 Build 22270

This release provides the following fixes and enhancements:

- Added a new command to identify a device by blinking its LEDs.
- Added support to display the Segment and Group Information of a RAID 10 Logical Device.
- Added a new UNINIT/TASK command to clear drive configuration for physical devices containing RAID Information Sector (RIS) that have been masked in order to protect the configuration data.
- Added the ability to show SGPIO virtual SEP information to show enclosure devices for SGPIO backplanes.
- Added support to report the array status in cases where a logical drive fails.
- Added the ability to display physical drive mount point information.
- Added the ability to display logical drive mount point information.

2.3 Limitations

This section details the limitations in this release.

2.3.1 Firmware Limitations

2.3.1.1 Limitations for Firmware Release 4.90 Build 0

There are no known limitations for this release.

2.3.1.2 Limitations for Firmware Release 4.80 Build 0

There are no known limitations for this release.

2.3.1.3 Limitations for Firmware Release 4.52 Build 0

There are no known limitations for this release.

2.3.1.4 Limitations for Firmware Release 4.50 Build 0

There are no known limitations for this release.

2.3.1.5 Limitations for Firmware Release 4.02 Build 0

There are no known limitations for this release.

2.3.1.6 Limitations for Firmware Release 3.52 Build 0

There are no known limitations for this release.

2.3.1.7 Limitations for Firmware Release 3.02 Build 0

This release includes the following limitation:

- When a large enclosure configuration (for example, 5 enclosures * 24 drives) is hot plugged, a few drives may go undetected. If the controller and the large enclosure configuration boot at the same time, there is no issue and all the drives get detected.

2.3.1.8 Limitations for Firmware Release 2.56 Build 0

There are no known limitations for this release.

2.3.1.9 Limitations for Firmware Release 2.50 Build 0

This release includes the following limitation:

- When maxView is running on controllers that have internal connectors under Windows operating systems, the controller logs contain spurious prints pointing to a command failure.
 - *Exposure:* All controllers that have internal connectors (8i, 8i8e, 4i4e, etc.) running Windows Server OS and MaxView.
 - *Workaround:* None

2.3.1.10 Limitations for Firmware Release 2.04 Build 0

This release includes the following limitations:

- I2C bus association in the "External Signal Group" does not take effect as it is not read by the tool. The tool uses the "I2C bus" field only for sake of firmware initialization. It is not used to assign the I2C bus to the specific connector. I2C buses 5,6,7, and 8 have fixed associations with External/Internal connectors 1,2,3, and 4.
- There is fixed association between the "External Signal Group x" and "External Connector x". The "Phy Map" field is expected to be same for both groups.

There are no known limitations for this release.

2.3.1.11 Limitations for Firmware Release 1.54 Build 0

This release includes the following limitations:

- When setting SAN policy to OnlineAll, the new setting will NOT bring the drives already discovered online; it will only change the policy. Even a reboot will not change the offline state for drives that are already discovered. Instead, this will have to be done manually per drive using the disk manager. Newly discovered drives from the point of configuration forward will automatically be shown as online by default.

Note:

Most data centers deploy images with the SAN policy=OnlineALL as part of their images before they install new systems (if they do not plan to hook them up in a shared SAS topology).

- A drive failed during discovery is not visible to host configuration tools..
- LED update behavior inconsistent on products based on LSI SAS-2 expanders. Root cause to the expander ignoring valid LED update requests.

2.3.1.12 Limitations for Firmware Release 1.02 Build 0

This release includes the following limitations:

- Chenbro RM43260-6G JBOD and AIC expanders (e.g., AIC XJ3000-4693s) are not supported in this release.
- RBODs are not supported in this release.
- If a physical disk is removed while the server is offline and that disk was a part of a logical volume undergoing transformation from a set of disks including this disk to another set of disks, the controller may lockup on the subsequent boot.

WORKAROUND: Avoid offline drive movement during volume transformation operations.

- The controller may hang if an attempt is made to configure a logical volume using disks which are in a spun-down state when in “mixed” connector mode.
 WORKAROUND: Spin the drive up prior to configuration since it is an HBA target at that time.
- LED update behavior inconsistent on products based on LSI SAS-2 expanders. Root cause to the expander ignoring valid LED update requests.

2.3.2 BIOS/UEFI Limitations**2.3.2.1 Fixes and Enhancements for UEFI/Legacy BIOS Build 1.2.6.3/1.2.6.4**

There are no changes for UEFI/HII Build 1.2.6.3 from the previous release.

There are no changes for Legacy BIOS Build 1.2.6.4 from the previous release.

2.3.2.2 Limitations for UEFI and Legacy BIOS Release 1.2.6.3

There are no known limitations for this release.

2.3.2.3 Limitations for UEFI/Legacy BIOS Build 1.2.5.2/1.2.5.4

There are no known limitations for this release.

2.3.2.4 Limitations for UEFI/Legacy BIOS Build 1.2.5.2/1.2.5.2

There are no known limitations for this release.

2.3.2.5 Limitations for UEFI/Legacy BIOS Build 1.2.5.1/1.2.5.2

There are no known limitations for this release.

2.3.2.6 Limitations for UEFI/Legacy BIOS Build 1.2.4.1252/1.2.4.1254

There are no known limitations for this release.

2.3.2.7 Limitations for UEFI/Legacy BIOS Build 1.2.3.1202/1.2.3.1206

There are no known limitations for this release.

2.3.2.8 Limitations for UEFI/Legacy BIOS Build 1.2.3.1202/1.2.3.1201

There are no known UEFI/HII limitations for this release.

This release includes the following Legacy BIOS limitations:

- Accessing the set boot drive menu in certain servers causes a memory corruption, and random hangs/screen corruption might be seen. This issue is seen mostly if the user sets a secondary boot drive without setting a primary boot drive.

WORKAROUND: Use OS tools to manage boot drives, if hang is seen from CTRL-A.
- In CTRL-A utility, there is no option to Initialize/Uninitialize the drives. When using a drive which has partition data, the user will see the following error/warning message "The backup GPT Data is located beyond the end of disk".

WORKAROUND: Remove all GPT partitions before using HDDs with this release.
- Firmware can crash if the attached SAS switch is reset multiple times in a short period of time.

2.3.2.9 Limitations for UEFI/Legacy BIOS Build 1.2.2.1107/1.2.2.1104

There are no known limitations for this release.

2.3.2.10 Limitations for UEFI/Legacy BIOS Build 1.2.0.1003

This release includes the following limitations.

- For CTRL-A and UEFI/HII:
 - Bay number-based sorting is not implemented. Drives will be sorted in phy order.
 - Drive identification operations are not supported in HBA mode.

2.3.3 Driver Limitations

2.3.3.1 Windows Driver Limitations

2.3.3.1.1 Limitations for Windows Driver Build 59002

There are no known limitations for this release.

2.3.3.1.2 Limitations for Windows Driver Build 58015

There are no known limitations for this release.

2.3.3.1.3 Limitations for Windows Driver Build 57011

There are no known limitations for this release.

2.3.3.1.4 Limitations for Windows Driver Build 56008

There are no known limitations for this release.

2.3.3.1.5 Limitations for Windows Driver Build 55022

There are no known limitations for this release.

2.3.3.1.6 Limitations for Windows Driver Build 54013

This release includes the following limitations:

- A BSOD failure can result from a DPC Watchdog Violation when multiple host IOs have timed out to HBA targets, RAID logical volumes, or when the controller firmware has crashed. When these events occur, the Windows OS will try to reset all the problematic targets, but a lock problem in the windows kernel can occur due to multiple threads executing a reset at the same time. This results in the DPC Watchdog Violation BSOD. Microsoft has reported that this problem is fixed for Windows Server 2012 R2 and Windows Server 2016.
 - The fix for Windows Server 2016 can be found here:
<http://www.catalog.update.microsoft.com/Search.aspx?q=%22Windows%20Server%202016%22%20Cumulative>
From the list of updates, select the one called "2017-06 Cumulative Update for Windows Server 2016 for x64-based Systems (KB4022715)", which has the fix.
 - The link to a Windows Server 2012 R2 fix is not known at the time of this document's release. Please check the Microsoft website for future Windows 2012 Server R2 cumulative updates.

2.3.3.1.7 Limitations for Windows Driver Build 52013

This release includes the following Windows limitation:

- In a configuration where the OS is installed on our controller and when the controller is crashed artificially under a heavy workload, the system will experience a BSOD (no known workaround).

2.3.3.1.8 Limitations for Windows Driver Build 50666

This release includes the following limitation:

- When an OS is installed on drives that cause IO timeouts, the controller may not boot to the OS properly because it is unable to read data from the drives. This can lead to continuous IO retries from the OS that result in a system hang.

WORKAROUND: Replace the drives with known operational drives.

2.3.3.1.9 Limitations for Windows Driver Build 50637

This release includes the following limitations:

- When an OS is installed on drives that cause IO timeouts, the controller may not boot to the OS properly because it is unable to read data from the drives. This can lead to continuous IO retries from the OS that result in a system hang.

WORKAROUND: Replace the drives with known operational drives.

2.3.3.2 Linux, VMware, and XenServer Driver Limitations

2.3.3.2.1 Limitations for Linux, VMware, XenServer Driver Build 59002

This release includes the following limitation:

- The AACRAID driver for Ubuntu 16.04.1 is built against the same kernel that is part of the ISO image of Ubuntu 16.04.1, But an updated kernel was provided by Ubuntu. The end customers/users may have to setup an auto-update on Ubuntu or manually update the kernel to the later version. To update

the AACRAID driver, use the DKMS tools, which allows the customers to update the driver to the installed version of the kernel.

2.3.3.2.2 Limitations for Linux, VMware, XenServer Driver Build 58012

There are no known limitations for this release.

2.3.3.2.3 Limitations for Linux, VMware, and XenServer Driver Build 57013

There are no known limitations for this release.

2.3.3.2.4 Limitations for Linux and XenServer Build 56008 and VMware Driver Build 56009

There are no known limitations for this release.

2.3.3.2.5 Limitations for Linux, VMware, and XenServer Driver Build 55022

There are no known limitations for this release.

2.3.3.2.6 Limitations for Linux, VMware, and XenServer Driver Build 54013

This release includes the following XenServer limitations:

- There is a bug in the XenServer OS which prevents certification from passing. Customers running XenServer 7.0 should apply hotfix XS70E030, which can be found at the following link: <https://support.citrix.com/article/CTX223825>

There are no known Linux or VMware limitations for this release.

2.3.3.2.7 Limitations for Linux and XenServer Build 53005 and VMware Driver Build 53004

There are no known limitations for this release.

2.3.3.2.8 Limitations for Linux, VMware, and XenServer Driver Build 52011

This release includes the following Linux, VMware, and XenServer limitations:

- A PSOD error occurs whenever the controller firmware is locked up and when the driver tries to reset the adapter.
- HBA 1000 driver v1.2.1-B5xxxx does not discover enclosure, but the inbox driver version 1.2.0-B30300 does (no workarounds are available).
- `ls SCSI -t` command is not supported for retrieving the SAS addresses of the attached targets, as the SAS transport layer support is missing in the AACRAID driver. This limitation exists starting with the first release of the product.
- If the controller firmware locks up due to misbehaving targets or for any other reason, the driver attempts to reset the controller firmware. In some scenarios, the recovery of the controller does not happen, as the driver fails to re-initialize the controller firmware.

2.3.3.2.9 Limitations for Linux Driver Build 50663

This release includes the following Linux limitations:

- Linux drivers hang on hot-plug of drives during I/O load.
- In SuSE 11, SP3 (and RHEL/CentOS 6.4) (32-bit and 64-bit), the inbox driver takes precedence over third-party drivers. As a result, the OS loads the inbox driver rather than installing from the out-of-box driver. (Note that RHEL/CentOS 6.4 is included in the release package as-is, without support.)

WORKAROUND: The `aacraid` module must be added to the blacklist during installation for both 32- and 64-bit out-of-box drivers as follows:

- For SuSE 11, SP3:

1. Boot from the SuSE installation DVD.
2. Add 'brokenmodules=aacraid dud=1' to the boot menu.
3. Press the F6 key to update the driver during installation.
4. Insert the USB driver disk.
5. On the 'Welcome screen', press CTRL+ALT+F2 to switch to the console and then type the following commands:

```
# insmod ./update/000/modules/aacraid.ko
# rm -rf /etc/modprobe.d/blacklist /etc/modprobe.d/noload
```

6. Press CTRL+ALT+F7 to return to the installation screen and continue the installation.
7. When the installation is 60-90% complete, press CTRL+ALT+F2 to switch back to the console.
8. Type the command:

```
# cp -a /update/install /mnt/tmp
```

9. Press CTRL+ALT+F7 to return to the installation screen and complete the installation.
- For RHEL/CentOS 6.4:
 1. Add "blacklist=aacraid linux dd" to the Boot menu.
 2. When prompted to "Please choose the Driver Update medium", highlight the USB partition or CD Drive medium and then click OK.
 3. On the Welcome screen, press CTRL+ALT+F2 to switch to the console.
 4. Type this command:

```
Remove the entry "blacklist aacraid" from /etc/modprobe.d/anaconda.conf modprobe
aacraid
```

5. Press CTRL+ALT+F7 to return to the Welcome screen.
 6. Complete the installation following the on-screen instructions
- SLES 10 SP4 installation hangs with the inbox driver. A fix will be available for a future release.

2.3.3.2.10 Limitations for Linux Driver Build 50639

This release includes the following limitations:

- The Linux OS may hang during boot if the BIOS for the HBA 1000 Series adapter has been loaded before an Microsemi Adaptec RAID controller (ARC) adapter (Series 6/7/8) in the same system.
 WORKAROUND: Swap the two controllers on the chassis to load the BIOS for the ARC product before the BIOS for the HBA 1000 Series adapter.
- The Linux OS does not boot if there are two or more ARC products in the system along with the HBA 1000 Series adapter.

2.3.3.2.11 Limitations for VMware Driver Build 50663

This release includes the following limitations:

- An error handler-triggered reset could potentially cause a firmware crash with the current release driver.

2.3.3.2.12 Limitations for VMware Driver Build B50639B50629

This release includes the following limitation:

- ESXi 6 installation fails with a PSOD when the HBA 1000 Series adapter and an Microsemi Adaptec Series 7 controller are present in the system.

WORKAROUND: Swap the two controllers on the chassis to load the BIOS for the ARC product before the BIOS for the HBA 1000 Series adapter.

2.3.3.2.13 Limitations for XenServer Driver Build 50663

This release includes the following Citrix XenServer limitation:

- An error handler-triggered reset could potentially cause a firmware crash with the current release driver.
- XenServer 6.5 fails to install when using a USB drive for driver installation.

WORKAROUND: To install the provided controller driver with Citrix XenServer, you must burn the XenServer driver iso image to a CD or DVD. You will be prompted to insert the XenServer driver CD/DVD twice. You will need the XenServer installation CD to complete this task. You must have administrator privileges to install the driver image.

2.3.3.3 General Limitations for Driver Build 50659

This release includes the following general limitation:

- The SAS address of drives cannot be retrieved using the "lsscsi -t" command.

2.3.4 Management Software Limitations

2.3.4.1 Limitations for maxView Storage Manager/ARCCONF Version 3.02.00 Build 23699

There are no known limitations for this release.

2.3.4.2 Limitations for maxView Storage Manage/ARCCONF Version 3.02.00 Build 23600

This release includes the following limitation:

- Drive firmware update will not be supported for Intel SSD 3510 model.

2.3.4.3 Limitations for maxView Storage Manage/ARCCONF Version 3.00.00 Build 23488

There are no known limitations for this release.

2.3.4.4 Limitations for maxView Storage Manage/ARCCONF Version 2.06.00 Build 23164

There are no known limitations for this release.

2.3.4.5 Limitations for maxView Storage Manage/ARCCONF Version 2.05.00 Build 22932

There are no known limitations for this release.

2.3.4.6 Limitations for maxView Storage Manage/ARCCONF Version 2.04 Build 22665

There are no known limitations for this release.

2.3.4.7 Limitations for maxView Storage Manager/ARCCONF Version 2.03 Build 22476

There are no known limitations for this release.

2.3.4.8 Limitations for maxView Storage Manager/ARCCONF Version 2.02 Build 22404

There are no known limitations for this release.

2.3.4.9 Limitations for maxView Storage Manager/ARCCONF Version 2.01.00 Build 22270

This release includes the following limitations:

- The ARCCONF SEEPROM UPDATE command is not supported for this product.

Refer to the *README: maxView Storage Manager & ARCCONF Command Line Utility 5* for a full list of known issues applicable to Smart-Family products using the maxView Storage Manager and ARCCONF utilities.

2.3.4.10 Limitations for maxView Storage Manager/ARCCONF Version 2.00.00 Build 21811

This release includes the following limitations specific to the HBA 1000 Series adapter:

- SES passthrough commands to expanders are failing.
- The Task ID of a secure erase changes in the maxView Storage Manager or the ARCCONF CLI while the task is in progress. There is no workaround for this because Smart-firmware-based products have no concept of task scheduling or task IDs.

Refer to the *README: maxView Storage Manager & ARCCONF Command Line Utility 5* for a full list of known issues applicable to Smart-Family products using the maxView Storage Manager and ARCCONF utilities.

3 Implementing a Host Application for a Bus Management Controller

As part of the firmware release package, Microsemi provides PBSI reference code for implementing a host application for a Bus Management Controller (BMC) to access details about this product and devices attached to it.

The reference files include examples of the commands shown in [Table 3 • PBSI Commands](#) on page 54.

Table 3 • PBSI Commands

Command	Description
get_adapter_handles	Displays the adapter handles
get_adapter_info	Displays the adapter information
get_version_info	Displays the PBSI and DST version information
get_adap_vital_stats	Displays the vital adapter information
get_gb_info	Displays the green backup unit information
get_topology_info	Displays the adapter topology information

To compile the PBSI application:

1. Get the appropriate I2C libraries.
2. Edit the `host_interfacing.c` file to use the platform-specific APIs.
3. Edit the makefile to use the appropriate toolchain for compilation.

**Microsemi**

2355 W. Chandler Blvd.
 Chandler, AZ 85224 USA

Within the USA: +1 (480) 792-7200
 Fax: +1 (480) 792-7277

www.microsemi.com © 2020 Microsemi and its corporate affiliates. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation and its corporate affiliates. All other trademarks and service marks are the property of their respective owners.

Microsemi's product warranty is set forth in Microsemi's Sales Order Terms and Conditions. Information contained in this publication is provided for the sole purpose of designing with and using Microsemi products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is your responsibility to ensure that your application meets with your specifications. THIS INFORMATION IS PROVIDED "AS IS." MICROSEMI MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROSEMI BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE WHATSOEVER RELATED TO THIS INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROSEMI HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROSEMI'S TOTAL LIABILITY ON ALL CLAIMS IN RELATED TO THIS INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, YOU PAID DIRECTLY TO MICROSEMI FOR THIS INFORMATION. Use of Microsemi devices in life support, mission-critical equipment or applications, and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend and indemnify Microsemi from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microsemi intellectual property rights unless otherwise stated.

Microsemi Corporation, a subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), and its corporate affiliates are leading providers of smart, connected and secure embedded control solutions. Their easy-to-use development tools and comprehensive product portfolio enable customers to create optimal designs which reduce risk while lowering total system cost and time to market. These solutions serve more than 120,000 customers across the industrial, automotive, consumer, aerospace and defense, communications and computing markets. Headquartered in Chandler, Arizona, the company offers outstanding technical support along with dependable delivery and quality. Learn more at www.microsemi.com.

PMC-2152169