Release Notes

HBA 1100 Software/Firmware

Released
June 2018
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Revision Date</th>
<th>Details of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>June 2018</td>
<td>SR2.3 Production Release</td>
</tr>
<tr>
<td>9</td>
<td>October 2017</td>
<td>Update Supported OSs</td>
</tr>
<tr>
<td>8</td>
<td>October 2017</td>
<td>First Production Release</td>
</tr>
<tr>
<td>1-7</td>
<td>December 2017 - July 2017</td>
<td>Pre-Production Releases</td>
</tr>
</tbody>
</table>
Contents

Software License Agreement...........................................................................................................5

1 About This Release.......................................................................................................................8
  1.1 Release Identification................................................................................................................8
  1.2 Components and Documents Included in this Release...........................................................8
  1.3 Files Included in this Release..................................................................................................9

2 What is New?.............................................................................................................................12
  2.1 Features....................................................................................................................................12
  2.2 Fixes........................................................................................................................................12
    2.2.1 Firmware Fixes......................................................................................................................12
    2.2.1.1 Fixes and Enhancements for Firmware Release 1.60 B0.....................................................12
    2.2.2 UEFI Fixes............................................................................................................................17
    2.2.2.1 Fixes and Enhancements for UEFI Build 1.3.4.11/Legacy BIOS Build 1.3.4.6...............17
    2.2.3 Driver Fixes........................................................................................................................20
    2.2.3.1 Fixes and Enhancements for Linux Driver Build 1.1.4-132..............................................19
    2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 1.0.2-1038.....................................19
    2.2.3.3 Fixes and Enhancements for Solaris Driver Build 1.0.2-1038........................................20
    2.2.3.4 Fixes and Enhancements for Windows Builds 100.64.2.64 and 6.64.2.64 Pass 1...........20
    2.2.3.5 Fixes and Enhancements for VMware Driver Build 1.0.2-1038.....................................21
    2.2.4 Management Software Fixes.............................................................................................21
    2.2.4.1 Fixes and Enhancements for Arcconf/maxView Build B23167.......................................21
  2.3 Limitations..........................................................................................................................26
    2.3.1 Firmware Limitations........................................................................................................26
    2.3.1.1 Limitations for Firmware Release 1.60 Build 0...............................................................26
    2.3.1.2 Limitations for Firmware Release 1.32 Build 0...............................................................26
    2.3.2 UEFI Limitations................................................................................................................26
    2.3.2.1 Limitations for UEFI Build 1.3.4.11/Legacy BIOS Build 1.3.4.6.................................26
    2.3.3 Driver Limitations...............................................................................................................26
    2.3.3.1 Limitations for Linux Driver Build 1.1.4-132.................................................................26
    2.3.3.2 Limitations for Windows Driver Builds 100.64.2.64 Pass 1 and 6.62.2.64 Pass 1.........26
    2.3.3.3 Limitations for FreeBSD Driver Build 1.0.2-1038........................................................26
    2.3.3.4 Limitations for Solaris Driver Build 1.0.2-1038............................................................27
    2.3.3.5 Limitations for VMware Driver Build 1.0.2-1038........................................................27
    2.3.3.6 Limitations for VMware Driver 1.0.0-1060..................................................................27
    2.3.4 Hardware Limitations.......................................................................................................27
    2.3.5 Management Software Limitations...................................................................................27
    2.3.5.1 Limitations for Arcconf and maxView Build B23167.....................................................27

3 Updating the Board Firmware for PQI Operation....................................................................28
  3.1 Updating Controllers to latest (PQI) Firmware.......................................................................28

4 Installing the Drivers.................................................................................................................29
## Tables

Table 1 • Release Summary .......................................................................................................................................8
Table 2 • Firmware Files ............................................................................................................................................9
Table 3 • Firmware Programming Tools ....................................................................................................................9
Table 4 • Windows Storport Miniport SmartPQI Drivers ..........................................................................................9
Table 5 • Linux SmartPQI Drivers ..............................................................................................................................9
Table 6 • FreeBSD, Solaris, and VMware SmartPQI Drivers ....................................................................................10
Table 7 • Host Management Utilities ......................................................................................................................10
Table 8 • Feature Summary .....................................................................................................................................12
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
1 About This Release

The development release described in this document includes firmware, OS drivers, tools, and host management software for the HBA 1100 Series adapter solutions from Microsemi.

1.1 Release Identification

The firmware, software, and driver versions for this release are shown in Table 1 • Release Summary

<table>
<thead>
<tr>
<th>Package Release Date</th>
<th>June 26, 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Version</td>
<td>1.60 B0 w/</td>
</tr>
<tr>
<td></td>
<td>Basecode 06.03.031.000</td>
</tr>
<tr>
<td>UEFI Version</td>
<td>1.3.4.11</td>
</tr>
<tr>
<td>Legacy BIOS</td>
<td>1.3.4.6</td>
</tr>
<tr>
<td>Driver Versions</td>
<td>Windows SmartPQI:</td>
</tr>
<tr>
<td></td>
<td>Windows 2012/2016: 100.64.2.64 Pass 1</td>
</tr>
<tr>
<td></td>
<td>Windows 2008: 6.64.2.64 Pass 1</td>
</tr>
<tr>
<td></td>
<td>Linux SmartPQI:</td>
</tr>
<tr>
<td></td>
<td>RHEL 6/RHEL 7/RHEL 7.5/SLES 11/SLES 12/SLES 15: 1.1.4-132</td>
</tr>
<tr>
<td></td>
<td>Ubuntu 14/16: 1.1.4-132</td>
</tr>
<tr>
<td></td>
<td>Debian 8/9: 1.1.4-132</td>
</tr>
<tr>
<td></td>
<td>Oracle Linux 7: 1.1.4-132</td>
</tr>
<tr>
<td></td>
<td>Citrix xenServer 7: 1.1.4-132</td>
</tr>
<tr>
<td></td>
<td>VMware SmartPQI:</td>
</tr>
<tr>
<td></td>
<td>vSphere 6: 1.0.2-1038</td>
</tr>
<tr>
<td></td>
<td>FreeBSD/Solaris SmartPQI:</td>
</tr>
<tr>
<td></td>
<td>FreeBSD 10/11: 1.0.2-1038</td>
</tr>
<tr>
<td></td>
<td>Solaris 11: 1.0.2-1038</td>
</tr>
</tbody>
</table>

arcconf/Maxview            B23167

1.2 Components and Documents Included in this Release

Download the firmware, drivers, host management software, and supporting documentation for your HBA1100 controller solutions from the Microsemi Web site at https://start.microsemi.com.
1.3 Files Included in this Release

This release consists of the files listed in the following tables:

Firmware Files

Table 2 • Firmware Files

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Pre-Assembly Use</th>
<th>Post-Assembly Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmartFWx100.bin</td>
<td>Programmable NOR Flash File. Use to program NOR Flash for boards that are already running firmware.</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SmartFWx100_v1.29_b314.bin</td>
<td>Programmable NOR Flash File. Use to program NOR Flash for boards that are already running firmware.</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 3 • Firmware Programming Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Executable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arconf romupdate</td>
<td>The command allows to upgrade/downgrade the firmware and BIOS image to the controller.</td>
<td>Refer to Table 7 • Host Management Utilities</td>
</tr>
<tr>
<td>maxView firmware upgrade wizard</td>
<td>The firmware upgrade wizard allows to upgrade/downgrade the firmware and BIOS image to one or more controller(s) of same model in the system.</td>
<td>Refer to Table 7 • Host Management Utilities</td>
</tr>
</tbody>
</table>

Driver Files

Table 4 • Windows Storport Miniport SmartPQI Drivers

<table>
<thead>
<tr>
<th>Package</th>
<th>Drivers</th>
<th>Binary</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>Server 2016 and Windows 10, Server 2012, R2 and Windows 8.1, 8</td>
<td>SmartPqi.sys</td>
<td>x64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SmartPqi.inf</td>
<td>x64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SmartPqi.cat</td>
<td>x64</td>
</tr>
<tr>
<td>2008</td>
<td>Server 2008, 2008 R2 and Windows 7</td>
<td>SmartPqi.sys</td>
<td>x64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SmartPqi.inf</td>
<td>x64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SmartPqi.cat</td>
<td>x64</td>
</tr>
</tbody>
</table>

Table 5 • Linux SmartPQI Drivers

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Intel/AMD x64</th>
<th>Cavium ThunderX2 ARM x64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux 7.5¹</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 7.4², 7.3, 7.2, 7.1, 7.0</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Drivers | Intel/AMD x64 | Cavium ThunderX2 ARM x64
--- | --- | ---
Red Hat Enterprise Linux 6.9, 6.8, 6.7, 6.6 | X | 
SuSE Linux Enterprise Server 15 | X | X
SuSE Linux Enterprise Server 12 SP3 | X\(^1\) | X\(^2\)
SuSE Linux Enterprise Server 12, SP2, SP1, Base | X\(^1\) | 
SuSE Linux Enterprise Server 11 SP4, SP3 | X\(^1\) | 
Oracle Linux 7.4, 7.3, 7.2 | X | 
Oracle Linux 7.3 with UEK 4.1.12-61.1.18 | X | 
Oracle Linux 7.2 with UEK 3.10.0-327.el7 | X | 
Ubuntu 16.04.4 LTS | X | X
Debian 9.x | X | 
Debian 8.9 | X | 
CentOS 7.5\(^1\) | X | X
Citrix xenServer 7.1 | X | 

**Note:**

1. To mitigate against the Spectre Variant 2 vulnerability, the RHEL 6.9, RHEL 7.4, RHEL 7.5 and SLES11 SP3 and higher drivers have been compiled to avoid the usage of indirect jumps. This method is known as “Retpoline”.

2. SLES 12 SP3 cannot be installed on drives attached to the HBA 1100 controller due to a CPU and OS installation issue. This driver will support the HBA 1100 controller for non-boot drives.

### Table 6 • FreeBSD, Solaris, and VMware SmartPQI Drivers

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreeBSD 11.0, 10.3, 10.2</td>
<td>x64</td>
</tr>
<tr>
<td>Solaris 11.3</td>
<td>x64</td>
</tr>
<tr>
<td>VMware 6.0, 6.5, 6.7</td>
<td>x64</td>
</tr>
</tbody>
</table>

### Host Management Software

### Table 7 • Host Management Utilities

<table>
<thead>
<tr>
<th>Description</th>
<th>OS</th>
<th>Executable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCCONF Command Line Utility</td>
<td>Windows x64&lt;br&gt;Linux x64&lt;br&gt;VMware EXSi 5.5/6.0&lt;br&gt;XenServer 6.5.1</td>
<td>See the Arcconf download package for the OS-applicable installation executable.</td>
</tr>
<tr>
<td>Description</td>
<td>OS</td>
<td>Executable</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARCCONF for UEFI</td>
<td>Windows x64</td>
<td>See the firmware download package applicable for the controller.</td>
</tr>
<tr>
<td></td>
<td>Linux x64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VMware ESXi 5.5/6.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XenServer 6.5.1</td>
<td></td>
</tr>
<tr>
<td>maxView Storage Manager</td>
<td>vCenter 5.5 and 6.0</td>
<td>See the VMware maxView Storage Manager download package for the OS-applicable installation executable.</td>
</tr>
<tr>
<td>maxView vSphere Plugin</td>
<td>Linux x64</td>
<td>See the maxView BootUSB download package for the .iso file.</td>
</tr>
<tr>
<td>Boot USB (offline or pre-boot) for ARCCONF and maxView Storage Manager</td>
<td>Linux x64</td>
<td></td>
</tr>
</tbody>
</table>
2 What is New?

2.1 Features

Table 8 • Feature Summary lists features supported for this release. Features new to this release are designated as "New".

Table 8 • Feature Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>Supported in this Release</th>
<th>Future Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>UEFI Driver, Boot Support</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Legacy Boot Support</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dynamic Power Management</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SMR Drive Support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMR Drive Support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enumeration, Unrestricted Command Flow-Through</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SATL Translation for HA/HM SMR Management</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Identify All Drive Types</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Driver Support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Linux</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>VMware</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FreeBSD</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Solaris</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OS certification</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Flash Support</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arcconf romupdate</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Microsoft Storage Spaces for SGPIO-based backplanes</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Arcconf Tool Support</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>maxView Tool Support</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

2.2 Fixes

2.2.1 Firmware Fixes

2.2.1.1 Fixes and Enhancements for Firmware Release 1.60 B0

This release includes the following fixes and enhancements:

- Fixed a controller deadlock when a disk is very slow during volume transformation.
  
  *Root Cause*: A slow drive had caused the Flexible Latency Scheduling (FLS) to kick in and throttle a drive down to one outstanding command that took a long time to complete. On completion of this last request with error, the firmware was making a determination to retry the original host operation but did not prompt other waiting disk requests to resume submission to the drive. Meanwhile, a volume transformation request
holding a stripe lock was stuck in the elevator and at least one host I/O was blocked on this stripe lock, causing the host to freeze.

- **Fix:** Corrected the disk request completion path to always prompt the elevator queue for new disk requests, regardless of the retry decision for the previous completion. In this case, the disk now has zero outstanding requests so FLS metrics can also reset and allow more than one request to be submitted, if available.

- **Risk:** Medium

- Fixed an issue where issuing an SMP command to the controller virtual SES target would lead to a controller firmware hang or lockup.
  - **Root Cause:** Firmware was stuck in a loop trying to find a servicer for the SMP request.
  - **Fix:** Since SES devices are not valid targets for SMP requests, reject the command.
  - **Risk:** Low

- Fixed a potential deadlock in environments exercising SAS CSMI SMP command pass-through requests.
  - **Root Cause:** The API was failing to release a request buffer when the operation was also being completed in the submission context.
  - **Fix:** Added a check to free the memory when in this context.
  - **Risk:** Low

- Fixed a potential controller lockup resulting from hot-removing and adding different JBODs online.
  - **Root Cause:** A data structure tracking JBOD bay information was not being freed properly during hot-removal.
  - **Fix:** Modified hot-removal case to properly free this memory.
  - **Risk:** Low

- Fixed an issue where firmware attempting to enable drive write-cache would not be successful with some drive models.
  - **Root Cause:** Some drive models do not support changing certain drive write-cache parameters, but the SCSI ModeSense request used by firmware to determine if the drive write-cache is changeable or not was not actually using the CHANGEABLE parameter in its discovery request. This would lead firmware to believe it could change something that it cannot.
  - **Fix:** Use the CHANGEABLE parameter in discovery, and then disallow modifying drive write-cache parameters that the drive says are not changeable.
  - **Risk:** Medium

- Fixed an issue where the activity LED for a drive would blink when other drives are undergoing a physical drive erase process.
  - **Root Cause:** Status checking code was incorrectly looping through all drives to determine readiness by sending SCSI TestUnitReady irrespective of actual configuration state. This would lead to the LED blinking even though some of those targets aren't actually processing IO's.
  - **Fix:** Limited scope of this readiness check only to drives relevant to RAID metadata process state checks.
  - **Risk:** Low

- Fixed a controller 0x1ABD lockup during controller boot caused by a deadlock between the DMA hardware self-test and parallel SAS topology discovery operations.
  - **Root Cause:** DMA requests associated with the self-test poll for completion in a tight loop, however the parallel discovery threads also wait on DMAs associated with data transfers for SCSI traffic to and from targets. The thread context running the self-test is higher priority than these other processes, so locking was becoming blocked when the SAS traffic was waiting on
• A previous DMA completion to be handled by a discovery thread that could never take place due to the CPU being held in the tight loop.
  • Fix: Modified the DMA self-test to yield the CPU periodically while waiting for the queued DMA operations to be completed so that other processes can continue to run.
  • Risk: Low

• Fixed an issue where the "last failure reason" for a drive was showing a stale value after a sanitize operation.
  • Root Cause: This value was not being cleaned up properly after the sanitize state machine was completed.
  • Fix: Re-initialize both the last and initial failure reasons when sanitize has completed.
  • Risk: Low

• Fixed an issue where SATA maximum queue depth was limited to 31.
  • Root Cause: Firmware was incorrectly subtracting one from the T13-specified max default NCQ queue depth of 32, and was also subtracting one from the queue depth reported in word 75 of the ATA device data.
  • Fix: Modified the behavior to match the T13 specification.
  • Risk: Low

• Fixed a controller lockup 0x20227 during boot.
  • Root Cause: The controller DMA self-test included a test that was allocating memory for a first test, initializing it, doing the test, freeing the memory, then allocating memory, and not initializing that memory before preforming the next test.
  • Fix: The intent appeared to have been to re-use the memory from the first test, but the actual code was freeing and releasing resources between the two tests. To fix, code was adjusted to re-use the original memory and to not free it until the testing is finished.
  • Risk: Low

• Fixed a controller 0x1E30 lockup where the PBSI functionality is allowed but is currently not enabled.
  • Root Cause: Effecting various configuration and state changes to the controller would generate PBSI event log entries. However, that event queue is not allocated when the feature is not enabled.
  • Fix: Added a filter to discard logging PBSI events if the feature is not enabled.
  • Risk: Low

• Added support for new JBODs.

• Fixed a firmware ASSERT resulting from an issue where controller firmware would free a buffer context for an I/O; then, attempt to use that buffer context only to discover that it is no longer a valid I/O context.
  • Root Cause: Firmware would queue the buffer context for delayed processing, followed quickly by a SATA drive reset or removal. The drive reset or removal would cause controller firmware to recycle the buffer context including modifying a field in the buffer from a request type to a response type. However, a pointer to this context remains queued for delayed processing. Subsequently, firmware dequeued this context and ASSERTed when the context was found to be invalid (response rather than request).
  • Fix: Modified the firmware to simply ignore the buffer context if the above mentioned field of request had been changed from request to response type. The firmware should not assert in this case because the response type field here indicates that the I/O has been flushed and the buffer context has been recycled. It is safe to ignore the queue entry here.
  • Risk: Low
• Fixed controller lockup 0x1BE0 resulting from perceived double-completion event of a single SCSI request.
  • **Root Cause:** This was a race condition in the firmware between checking a value and an atomic decrement, where another process modified the value between these two operations.
  • **Fix:** Save the result of the atomic decrement separately, and check against that result instead. Also increased the amount of debug information available if this situation occurs again.
  • **Risk:** Low

• Fixed UART print decode errors where SCSI error responses for 16-byte CDB’s are not correct.
  • **Root Cause:** SCSI error responses for 16-byte CDB’s which are not reads or writes were incorrectly being decoded using a 10-byte CDB layout.
  • **Fix:** Updating the decode logic to check the opcode and use the correct CDB layout.
  • **Risk:** Low

• Fixed a controller lockup 0xFFFFF001 when attempting to capture diagnostic dump from the controller.
  • **Root Cause:** This was result of a null pointer exception encountered in the controller’s sensor monitoring layer due to firmware not enforcing proper bounds when the API to request capture of buffered UART output was called.
  • **Fix:** Firmware was updated to add proper bounds so that appropriate errors are returned in that path rather than cause buffer overrun conditions.
  • **Risk:** Low

• Fixed a controller 0x1E30 lockup when the drive target queue depth is being examined or modified.
  • **Root Cause:** Firmware implemented an incorrect check to configure the target queue depth.
  • **Fix:** Incorrect debug check was removed.
  • **Risk:** Low

• Fixed a controller 0x27006 lockup when the drive target queue depth is being modified.
  • **Root Cause:** Firmware was incorrectly making the configuration API call while holding a spinlock which lead to deadlocks.
  • **Fix:** Changes were made in firmware to make these configuration changes while managing the spinlocks appropriately or to defer this activity to an appropriate context.
  • **Risk:** Medium

• Fixed an issue where firmware could deadlock resulting in 0x27006 lockup when a drive failed.
  • **Root Cause:** Firmware implemented a drive failure logic in the command processing path instead of implementing it in a hotplug thread causing IO processing to be halted waiting on itself.
  • **Fix:** Modified the logic attempting to save the device failure information to check and wait for the drive reset to complete before proceeding with submission since IO would have already halted.
  • **Risk:** Low

• Fixed an issue where firmware could deadlock resulting in 0x27006 lockup on spurious expander or JBOD power cycle.
  • **Root Cause:** Heartbeat and IO timeout processing context was attempting to queue a blocking SCSI request to spin down a failed target, but this activity is not supposed to be performed in that execution context.
  • **Fix:** Avoid SCSI activity associated with marking a device as failed while in the heartbeat context.
  • **Risk:** Low
• Fixed a controller 0x1E30 lockup after powering off an enclosure while a drive sanitize erase in progress.
  • Root Cause: Firmware had incorrectly implemented a debug trap looking for conditions where use of spinlock is incorrect.
  • Fix: These firmware checks were found to not actually provide any value since the mechanism used did not require spinlock protections, so these traps were removed.
  • Risk: Low

• Fixed the STP Resource Busy lockup issue.
  • Root Cause: The lockup was due to firmware being blocked from processing since the module responsible for handling STP resource ran out of buffer leading to completions not returned on time.
  • Fix: Firmware function that is responsible for fetching buffer for STP resource from the buffer pool has been changed to a non-blocking function call. Further firmware port event manager module was changed to release the buffer sooner to reduce the probability of exhausting this buffer pool.
  • Risk: Low

• Fix the selection of the hardware engine performing pattern fill and pattern check operations.
  • Root Cause: This was due to an issue in how firmware filled out the descriptor for the pattern fill and check operations which causes the hardware to always use engine 0 out of 3 engines. This is a problem when there are multiple engines in use because the issue will cause all requests to be directed to engine 0. This results in incorrect pattern fill and check results.
  • Fix: Firmware was modified to fill out the hardware descriptor correctly to ensure that the pattern fill and check operations are directed to the right engine.
  • Risk: Low

• Fix an I/O freeze timeout because device(s) are not removed during a port down event.
  • Root Cause: The discovery mechanism does not remove all devices from a port that has gone down if the port’s previous discovery instance was interrupted before re-discovering the whole port. This is incorrect behavior. The following sequence of events caused the problem:
    1. Discovery instance N runs, and marks all previously discovered devices to dirty. When discovery instance N finds the device, it’ll be marked as non-dirty.
    2. Discovery instance N gets interrupted and ends before all devices marked dirty are found again.
    3. Port goes down
    4. Discovery instance N+1 runs to remove all devices on the port, but it mistakenly only removes devices that are not dirty.
  • Exposure:
    • On expander port down, the removal of the leftover devices may be delayed as the devices will be granted Loss of Sync (LOS) first. The removal will be delayed until LOS expires.
    • On systems with multiple expanders going through port toggling, we may hit this particular case where the freed expander structure gets recycled to represent another expander with similar phy connections. The devices that were incorrectly not removed at port down will reference the now valid immunity of the reused expander structure and have its IO freeze extended which leads to reported IO freeze timeout.
  • Fix: Ensure that when a port goes down, all devices get marked for removal in the clean-up of the Discovery instance, such that they will be all be removed in the subsequent Discovery instance.
• Risk: Low

• Fixed an issue where tool-prompted drive LED updates are not always honored.
  • Root Cause: LED updates were not honored when multiple processes attempted to trigger an
    update simultaneously or if an update was already in progress.
  • Fix: Made changes to firmware implementation to track the processes attempting to update
    LED states and to check if an update is already in progress.
  • Risk: Low

• Fixed an issue where the bay number for a drive would be reported as "Unknown" after invoking
  the target re-query API.
  • Root Cause: This was due to firmware clearing the stored bay value prior to the target re-query
    operation but was not running the enclosure discovery code which sets this value.
  • Fix: Changes were made to not clear the bay value or other enclosure-specific values since the
    enclosure discovery is not being re-run for this operation.
  • Risk: Low

• Fixed an issue where firmware may become unresponsive while attempting to flash firmware or
  execute other RAID logical volume operations.
  • Root Cause: Previous commands processed by firmware can lead to future firmware operations
    accessing a stale value that is intended to be a device handle. Accessing this stale value causes
    a firmware thread to hang, which results in future host management software requests hanging
    or other RAiD operations (eg. rebuilds) not starting. This can lead to controller hangs, lockups,
    or OS failures if the OS is installed on drives behind the controller. For controllers with no
    drives attached that have a World Wide Name (WWN) that ends with the last byte being in
    the range from 0x80 to 0xFF, a firmware flashing operation requested by host management
    software will hang. This is due to the WWN being the stale value used by firmware that caused
    the thread to hang, which results in the firmware flash request to hang.
  • Fix: Firmware dereferences the correct structure pointer to get a valid device handle instead
    of a stale value.
  • Risk: Low

• Added capability to configure default drive write-cache settings across various target configurations
  (HDD, SSD), type (SAS, SATA), and use-case (configured, not configured).
• Modified the debug option to override target queue depths to be persistent through reboot.
• Suppressed some log messages related to backup power preservation for products that do not
  support this feature.

2.2.2 UEFI Fixes

Note: Microsoft signed and secure boot is supported.

2.2.2.1 Fixes and Enhancements for UEFI Build 1.3.4.11/Legacy BIOS Build 1.3.4.6

This release includes the following UEFI fixes and enhancements:

• Fixed an issue where a second attempt to Save Support Archive operation freezes the platform.
  • Root Cause: Close Protocol fails for the handle of hot removed file system device.
  • Fix: Close protocol to be called only for successfully opened handles.
  • Exposure: All previous versions.
  • Risk: Low.

• Fixed an issue where a increase in UEFI driver boot time.
• **Root Cause:** Platform continuously send Test Unit Ready command until it finds the success status on sense data which was not set as expected by UEFI driver Extended Scsi Pass thru protocol.
  
  • **Fix:** Set sense data in Extended SCSI Pass through protocol as per specification.
  
  • **Exposure:** All previous versions.
  
  • **Risk:** Low.

• Fixed an issue where the identification duration is not set to default value in Identify device.
  
  • Root Cause: HII tool was updated with previously set drive duration instead of the default value for drive blink duration.
  
  • Fix: Set to the default value for drive blink duration.
  
  • Exposure: All previous versions
  
  • Risk: Low

• Added an enhancement to support drive write cache control for non-configured drives.
  
  • Implementation: Added option under Controller Settings menu to configure drive write cache for non-configured drives.
  
  • Risk: Low

• Fixed an issue to Sanitize reports success when no sanitize occurs with API sanitize.
  
  • Root Cause: The message registry defined the message to take two parameters but software only provided 1 which prevented the malformed message from being generated.
  
  • Fix: Modify the message registry such that the message "CanErasePhysicalDrivePatternNotSupported" is defined to take 1 parameter. The new message string will now be: "Cannot erase physical drive %1 because the drive does not support the desired erase pattern".
  
  • Exposure: All previous releases
  
  • Risk: Low

• Fixed an issue where the identification duration is not set to default value in Identify device.
  
  • Root Cause: HII tool was updated with previously set drive duration instead of default value for drive blink duration.
  
  • Fix: Set to default value for drive blink duration.
  
  • Exposure: All previous versions
  
  • Risk: Low

• Fixed the missing PCIe segment field in Controller information menu.
  
  • Root Cause: PCI segment information is not provided.
  
  • Fix: Read the PCI segment value from EFI_PCI_IO_PROTOCOL_GET_LOCATION and the same information is being displayed in Controller Information menu.
  
  • Exposure: All previous versions.
  
  • Risk: Low.

• Fixed an issue where Erase Drive does not have option to select type of erase.
  
  • Root Cause: Erase pattern type is not provided as an option.
  
  • Fix: Erase type option has been added to choose from different erase patterns.
  
  • Exposure: All previous versions.
  
  • Risk: Low.

• Fixed an issue where the Driver Health Configuration required menu "Ignore" option should exit from driver health menu.
  
  • Root Cause: Driver health option to ignore does not perform any action.
• Fix: Ignore option has been removed.
• Exposure: All previous versions.
• Risk: Low.

• Fixed an issue where the system assert on disconnect controller handle from the EFI shell.
  • Root Cause: Freeing the ConfigAccess stack variable resulted in memory corruption and the system asserted.
  • Fix: Removed freeing ConfigAccess stack variable.
  • Exposure: All previous versions.
  • Risk: Low.

• Fixed an issue where the platforms that are having two byte slot IDs are shown incorrect in controller information menu.
  • Root Cause: Slot IDs are processed as one byte value.
  • Fix: UEFI driver to consider higher data type to accommodate bigger slot numbers and report to host wellness buffer.
  • Exposure: All previous versions.
  • Risk: Low.

2.2.3 Driver Fixes

2.2.3.1 Fixes and Enhancements for Linux Driver Build 1.1.4-132
This release includes the following fixes and enhancements:

• Added support for SLES15 and RHEL7.5.
• Improved the "driver busy" handling for synchronous requests.
  • Root Cause: A deadlock between the driver and firmware caused by a combination of the firmware failing to complete a synchronous request because it is waiting on the driver to send a LUN reset and the driver is assuming that the firmware would eventually complete a synchronous request.
  • Fix: Improved “driver busy” handling for synchronous requests.
  • Risk: Low
  • Exposure: The driver behavior has been the same since the out-of-box version of version of SmartPQI was introduced. The probability of this issue occurring is significantly low because it requires a number of factors (SATA drive with outstanding NCQ commands queued to it to suddenly stop responding at the exact same time that the failing drive has an INQUIRY command outstanding to it that was internally generated by the driver) to align perfectly to cause the conditions that expose the issue.

• Fixed an issue with deleting PQI queue.
  • Root Cause: In case of PQI Queue Delete commands (Queue Delete command for deleting InBound and OutBound queues) two same responses were sent to the driver by the firmware. The driver was only expecting one response.
  • Fix: The driver skips deleting PQI operational queues when there is an error creating a new queue group. It is not necessary to delete the queues because they get deleted during the PQI reset that is part of the error recovery path.
  • Risk: Low
  • Exposure: Unknown

2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 1.0.2-1038
This release includes the following fixes and enhancements:

• Removed OS_SLEEP form administrator queue request.
• **Root Cause:** Earlier firmware was taking time to write response IU that is why driver was waiting for 20 millisecond to complete response IU.
  • **Fix:** Firmware added the fix for it so that response will be written properly without waiting in driver side.

• Introduced new log level for discovery.
  • **Root Cause:** There was only one log level for initialization and driver discovery.
  • **Fix:** Added two separate log level DBG_INIT & DBG_DISC for debugging initialization and discovery part.

• Added OS specific macro to check request type.
  • **Root Cause:** Driver needs to check whether a pending request is an internal request or a Command sent by the upper layer.
  • **Fix:** Added OS specific macro to check the request type.

• Changed the bus transfer speed.
  • **Root Cause:** The bus transfer speed was mentioned was mentioned incorrectly.
  • **Fix:** Changed the bus transfer speed from 300000 kb/sec to 1200000 kb/sec.

• Removed the device linked list changes.
  • **Root Cause:** Driver is not using linked list implementation for maintaining device list.
  • **Fix:** To maintain the device list driver is using array implementation.

• Added Legacy interrupt support.
  • **Root Cause:** If system fails to allocate interrupt type MSIX. Smart controller will not be initialized and driver return error status while loading.
  • **Fix:** If MSIX interrupt allocation is failed then driver will make use of interrupt type legacy.

### 2.2.3.3 Fixes and Enhancements for Solaris Driver Build 1.0.2-1038
There are no known enhancements or fixes for this release.

### 2.2.3.4 Fixes and Enhancements for Windows Builds 100.64.2.64 and 6.64.2.64 Pass 1
This release includes the following fixes and enhancements:

• Fixed an issue where Windows Server 2016 failover cluster validation test was failing.
  • **Root Cause:** The cluster validation test sends the SCSI command Persistent Reservation (0xF5). The SCSI Persistent Reservation (0xF5) command was failing because the host driver SmartPqi was not returning the correct SCSI request status SRB_STATUS_ERROR(0x04) when the hard drive correctly returned SCSISTAT_RESERVATION_CONFLICT(0x18) and sense data length of zero for a failed Persistent Reservation (0xF5) command.
  • **Fix:** The driver now returns the correct SCSI request status SRB_STATUS_ERROR(0x04) when a SCSISTAT_RESERVATION_CONFLICT(0x18) and sense data length of zero condition is returned upon a Persistent Reservation (0xF5) command failure.
  • **Exposure:** All SmartPQI versions prior to 100.62.0.64 are affected and will fail the Windows Server 2016 failover cluster validation test.
  • **Risk:** Low

• Fixed an issue where the SmartPQI driver was not displaying SCSI error Sense Data via the Windows “TraceView” utility.
  • **Root Cause:** Driver did not contain complete and well formatted trace statements for displaying error information for a non-debug version of the driver.
  • **Fix:** Added a new driver routine that contains trace statement to display error information.
  • **Exposure:** All SmartPQI versions prior to 100.62.0.64.
2.2.3.5 **Fixes and Enhancements for VMware Driver Build 1.0.2-1038**

This release provides the following fixes and enhancements:

- Fixed an issue where path failover is not working in VSAN multi-path configurations.
  - **Root Cause:** When a path is removed, the controller firmware returns PQI_AIO_STATUS_AIO_PATH_DISABLED error status for the pending IOs. Upon receiving this, driver was reporting VMK_SCSI_HOST_ERROR to PSA, and that will result in I/O retry through the same path. This might delay the path switching in dual paths configuration and vSAN reports disk failure.
  - **Fix:** Return the appropriate error code so that path failover will occur.

- Fixed an issue with legacy interrupt support.
  - **Root Cause:** During driver load or attach if system failed to allocate MSIx interrupts to the smart controller, driver will be returned with an error as no resources and it will not initialize the controller or driver will not be loaded.
  - **Fix:** If Interrupt type MSIx is not available, driver will try to allocate and use interrupt type legacy.

- Fixed the PSOD during reboot -f command in balance power mode in ESXi 6.7.
  - **Root Cause:** Detach and "device scan" running parallel during reboot and that caused this issue. During detach, driver was not waiting for any device discovery to get completed. If an event is generated by the firmware for a rescan during detach, rescan will end up in a page fault as the resources are freed by the detach(). Along with this, the "delay" during the shutdown is also fixed.
  - **Fix:** Wait for rescan to complete in shutdown(). Correct the order of uninitialization.

- Fixed an issue for Esxi 6.7 IOVP certification DeviceStateChange test failure.
  - **Details:** DeviceStateChange test invokes/tests the driver load (attach, start) and unload (quiesce, detach) entries and verifies the device unclaim, claim, and IO 'able states. Above operation is performed in a loop for several iterations. Upon analyzing this issue it is found that after few test iterations, test was failing at verifying the unclaim state during unload operation.
  - **Root Cause:** This failure is due to driver is performing cache flush with SHUTDOWN as an halt_event from the detach entry. It is invalid to do this from detach entry point.
  - **Fix:** Made the code changes to do cache flush with "CACHE_FLUSH_ONLY" event from detach() routine, as detach is just a driver unload and cache flush with "SHUTDOWN" event from quiesce() routine, as quiesce will be invoked on system shutdown.

- Fixed an issue in HBA mode, where driver is exposing only one target instead of two in case of externally attached JBODs in a multipath configurations.
  - **Root Cause:** In HBA mode, driver was assigning the same target ID for both of the targets/paths for drives connected in multipath. Due to this, the driver was reporting only one path to SCSI midlayer.
  - **Fix:** Implemented “target ID pool” (stack). Whenever a new PD is reported, target ID will be taken from this pool and driver will be pushing the target id back to the pool, when the device is freed.

2.2.4 **Management Software Fixes**

2.2.4.1 **Fixes and Enhancements for Arcconf/maxView Build B23167**

This release includes the following fixes for Arcconf/maxView build B23167:
• Fixed an issue where connector display view (CN0/CN1) for expander attached cases was not displaying the redundant paths.
  • Root cause: Add the multiple active path, where it was only added for the multiple redundant path.
  • Fix: Implemented the multiple active path with the info icon. Click the icon to show the multiple active path in the table.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an arccconf playconfig failure when physical device write cache state is enabled.
  • Root cause: Arcconf playconfig failure occurred when the global physical device write cache state is enabled and when logical devices do not exist.
  • Fix: Modified the code to configure global physical device write cache only after a logical device creation.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue regarding SAS error counter support in arcconf/maxView.
  • Root cause: Changes to the display were blocked for SAS devices.
  • Fix: Made changes to display the SAS device error counters.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue related to the incorrect slot information in maxView in the Dual-path mode.
  • Root cause: Comparison was made on connector name instead of the connector vector ID of the enclosure object.
  • Fix: Added the comparison based on connector vector ID to display the slot information when connected with the dual-path mode.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue where maxView login failed on reboot in Debian 9.3s.
  • Root cause: On reboot, agent service was not running. The `stor_agent` was not creating the symbolic link in `/etc/rc.d`.
  • Fix: Made changes to the header section of the agent service script to create a symbolic link in `/etc/rc.d`.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue where the add system operation was allowed in the standalone mode maxView.
  • Root cause: Add system ribbon icon was enabled when maxView was installed in standalone mode, as the check to disable based on the mode of installation was missing.
  • Fix: Made changes to disable the add system ribbon icon and not to render the auto discovery related properties as they are not applicable in standalone mode maxView.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue where the enable/disable auto discovery operation was allowed in standalone mode maxView.
  • Root cause: In maxView, the tool-tip for auto discovery property was set to “auto-discovery task runs in the background each time maxView storage manager is started” which was not as per the design.
  • Fix: Made changes in tool-tip display for auto discovery as “Auto discovery provides an option to discover the agent service automatically” to inform user about the functionality.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the delete remote system operation failed.
Root cause: "Delete remote system" operation was failing for systems that does not have a proper hostname set.
Fix: Made changes to use system's IP address as reference for "delete remote system" operation instead of hostname.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the maxView help/tooltip showed Auto Discovery was enabled by default.
Root cause: Proper tooltip for the Auto Discover property was missing.
Fix: Added proper tooltip for the Auto Discover property.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the requirement was to update the arcconf ROMUPDATE command help.
Root cause: Help strings had legacy optional commands that were not valid for newer controller models.
Fix: Removed the legacy optional commands for ROMUPDATE command.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the maxView help/tooltip showed that the sound alarm was enabled by default.
Root cause: The sound alarm property was not enabled by default.
Fix: The sound alarm property is enabled by default as mentioned in description of maxView help/tooltip.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the WARNING message that was not valid for command arcconf Identify with options "ALL" and "TIME".
Root cause: A invalid warning message was displayed when TIME option was given.
Fix: Added the proper check to display the warning message based on input command line options.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the “Sanitize block erase" pattern was not displayed for SSD-SATA physical device that is supported in maxView.
Root cause: While iterating over the list from xml configuration, a space after comma skipped the loop and missed listing the rest of the supported sanitize methods.
Fix: Trimmed the space in the switch case and added all the supported sanitize methods to discover the "Sanitize block erase" option for the SSD-SATA physical device.
Exposure: All previous releases.
Risk: Low

Fixed an issue in which the Jump icon was missing in state field of PD on which the sanitize task is running in maxview after reboot.
Root cause: The value for "Unique id" field for a hard device was blank, which was displayed as it is in the GUI.
Fix: Made proper changes to display the unique ID as "Not available" instead of blank.
• **Exposure**: All previous releases.
• **Risk**: Low

- Fixed an issue in which the GUI of maxView did not allow selection of multiple devices during drive firmware upgrade.
  - **Root cause**: The GUI was blocking multiple device selection during firmware upgrade.
  - **Fix**: Added condition to allow multiple device selection.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue in which the arcconf command to modify i2cClockSpeed with the same i2cAddress failed.
  - **Root cause**: arcconf blocks modifying of i2cClockSpeed when the i2cAddress of the controller is same as provided in the command.
  - **Fix**: Added code to change any of the three i2c parameters on controller level and provide a valid error message on failure.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue where the event monitor sendEmail script prompts for password and displays the password on the screen.
  - **Root cause**: sendEmail script displays the password provided as input on screen.
  - **Fix**: Remove echo before providing password and set it back to on after that.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue where the harddrive over temperature event does not display in the Events tab.
  - **Root cause**: The old and new values for the "currentTemperature" property was not updating properly resulting in failed comparison logic to generate events.
  - **Fix**: Added logic to allow only new "currentTemperature" to be updated to allow comparison logic to work for generating event.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue where the enterprise view scroll bar of maxView was not functioning properly in Microsoft edge browser.
  - **Root cause**: When the focus was on maxView's enterprise tree section in Microsoft edge browser, the scrollbar was not functioning properly using key strokes.
  - **Fix**: Made changes specific to the Microsoft edge browser, to render the scroll bar only when the enterprise tree extends beyond the margin and not on mouse hover event from browser.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue in which the requirement was to change the device interface strings from 'Sata Ssd' to 'SATA SSD' in the event message.
  - **Root cause**: The interface strings from down layer was in camel case.
  - **Fix**: Added a method to convert interface strings in upper case for displaying in events.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue in which the system name value was allowing blank spaces for Add system operation.
  - **Root cause**: Validation for the empty spaces was missing in the new system dialog.
  - **Fix**: Added the validation for the empty spaces for the new system dialog.
  - **Exposure**: All previous releases.
• **Risk:** Low
• Fixed an issue in which there was a need to enable/disable the Monitoring and Performance (MNP) events in.
  • **Root cause:** There was no option to enable/disable the logging of MNP events related to the physical devices.
  • **Fix:** Made changes to add an option in maxView to enable/disable the logging of MNP events related to the physical devices.
  • **Exposure:** All previous releases.
  • **Risk:** Low

• Fixed an issue in which no events were generated in maxView after temperature crossed the threshold value (87 °C).
  • **Root cause:** Earlier the threshold temperature of controller was 87. The threshold temperature of the controller needs to be updated to 97 °C as per the new controller model specification.
  • **Fix:** Updated controller threshold temperature to 97 °C to generate the temperature event.
  • **Exposure:** All previous releases.
  • **Risk:** Low

• Fixed an issue in which the maxView CIM server service was not running on Fedora 27 OS after the system reboot (sign in failed).
  • **Root cause:** After rebooting, the Fedora 27 system audit was blocking the stor_cimserver script from starting the service.
  • **Fix:** Added "generate policy" setting as "allow rules" to stor_cimserver for starting the service.
  • **Exposure:** All previous releases.
  • **Risk:** Low

• Fixed an issue in which the maxView was freezing during un-installation.
  • **Root cause:** Installer was issuing the stopCIMService command, while waiting for the CIMServer to stop.
  • **Fix:** Made changes to the logic in the installer to query for CIMserver status and terminate it if in waiting state.
  • **Exposure:** All previous releases.
  • **Risk:** Low

• Fixed an issue in which connector mode was not set for internal connectors using arconf.
  • **Root cause:** When a connector has wide path, then the error message displayed was not proper.
  • **Fix:** Added unavailable reason to display an error message.
  • **Exposure:** All previous releases.
  • **Risk:** Low

• Fixed an issue in which the SETPOWER command was not listed in arconf help.
  • **Root cause:** SETPOWER was missing in arconf help.
  • **Fix:** Implemented a method to display SETPOWER in arconf help based on feature.
  • **Exposure:** All previous releases.
  • **Risk:** Low

• Fixed an issue in which the maxView Advanced Statistics setting was not persistent across reboot.
  • **Root cause:** The warning and information tooltip was missing when advanced statistics was changed to enabled.
• **Fix**: Added the warning and information tooltip as "Advanced statistics will be defaulted to disabled state on system reboot. User needs to enable the advanced statistics on every reboot to enable and view the statistics data." when advanced statistics is set to enabled.

• **Exposure**: All previous releases.

• **Risk**: Low

• Fixed an issue in maxView where there was a wrong status on the smart tab for HGST on the 0xc2 attribute.

  • **Root cause**: The check must be based on the raw value of 0xC2 attribute and on the threshold value when it is more than zero.

  • **Fix**: The changes in check are made on the raw value to display correct status.

  • **Exposure**: All previous releases.

  • **Risk**: Low

2.3 Limitations

2.3.1 Firmware Limitations

2.3.1.1 Limitations for Firmware Release 1.60 Build 0

There are no known limitations for this release.

2.3.1.2 Limitations for Firmware Release 1.32 Build 0

• Firmware release 1.32b0 may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations.

  • **Description**: Refer to entry "Fixed an issue where firmware may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations" in the Firmware fixes section.

  • A fix for this issue is available in the 1.60 B0 firmware release. If a firmware flash failure is occurring, try the following workarounds:

    • **Workaround**: If there are no target devices (expanders or drives) attached to the controller, attach a target device to the controller and try the host management operation again.

    • **Workaround**: If the system is operating using UEFI, the HII tool can be used to flash the firmware to this release as outlined in the *Microsemi Adaptec HBA 1100 Series Host Bus Adapters Installation and User’s Guide*, appendix entry "Updating the HBA 1100 Controller Firmware".

    • **Workaround**: If there are target devices attached to the controller and this issue occurs or none of the workarounds can be used, contact Microsemi Support.

2.3.2 UEFI Limitations

2.3.2.1 Limitations for UEFI Build 1.3.4.11/Legacy BIOS Build 1.3.4.6

There are no known limitations for this release.

2.3.3 Driver Limitations

2.3.3.1 Limitations for Linux Driver Build 1.1.4-132

There are no known limitations for this release.

2.3.3.2 Limitations for Windows Driver Builds 100.64.2.64 Pass 1 and 6.62.2.64 Pass 1

There are no known limitations for this release.

2.3.3.3 Limitations for FreeBSD Driver Build 1.0.2-1038

There are no known limitations for this release.
2.3.3.4 Limitations for Solaris Driver Build 1.0.2-1038
There are no known limitations for this release.

2.3.3.5 Limitations for VMware Driver 1.0.2-1038
There are no known limitations for this release.

2.3.3.6 Limitations for VMware Driver 1.0.0-1060
This release contains the following VMware driver limitation:
• In a server, if the VMware OS was installed with customized image created through PowerCLI, driver upgrade/ removal (live-install) is not supported.

2.3.4 Hardware Limitations
This release includes the following hardware limitations:
• Two Wire Interface (TWI) address conflicts can cause system DDR memory to not be discovered.
  • Description: The HBA1100 boards include two TWI targets on the host-facing SMBUS interface with the following slave addresses:
    • 0xA0 – Field Replaceable Unit (FRU) SEEPROM
    • 0xDE – PBSI (default)
  According to the JEDEC specification, the default TWI addresses for the DDR SPD is 0xA0-0xAE (the spec uses 7 bit addressing which is 0x50-0x57). On platform system board designs with SMBUS wiring that has both PCIe slots and DDR slots shared on the same TWI bus, the TWI devices for the DDR and Smart controller are exposed to address conflicts which can result in the system memory not being discovered. The Smart controller PBSI interface defaults to a value of 0xDE (0x6F in 7-bit addressing) and is not a problem unless it is changed to an address that conflicts with the JEDEC defined values. The Smart controller FRU SEEPROM is hardwired to 0xA0.
  • Workaround: None available. If this issue is encountered, contact your Microsemi support engineer to determine the next steps for your system.
  • Performance with workaround: Not applicable
  • Performance without workaround: Not applicable

2.3.5 Management Software Limitations

2.3.5.1 Limitations for Arcconf and maxView Build B23167
There are no known limitations for this release.
3 Updating the Board Firmware for PQI Operation

This section describes how to update the board's firmware components to the latest release.

3.1 Updating Controllers to latest (PQI) Firmware

This procedure describes how to prepare your board to be programmed with the latest board PQI firmware.

Note: Complete these procedures exactly as described for proper functionality. If you do not follow all of the steps correctly, you could encounter unusual runtime behavior.

Flashing the board to the latest PQI firmware:

This section describes how to update all the firmware components on HBA 1100 Series adapter boards to the latest release.

If the controller is currently running 1.04 b0 firmware, follow these steps:

1. Mandatory: Flash the controller with the provided “SmartFWx100_v1.29_b314.bin” image with arcconf/maxView software.
2. Mandatory: Reboot the system to refresh all components.
3. Mandatory: Flash the target with the provided " SmartFWx100.bin" image with arcconf/maxView software.
4. Mandatory: Cold boot the system to refresh all components.

If the controller is currently running 1.32 b0 firmware, follow these steps:

1. Mandatory: Flash the target with the provided “SmartFWx100.bin” image with arcconf/maxView software.
   • If the arcconf/maxView software becomes unresponsive or hangs then power cycle the system to recover and refer to firmware limitation section Limitations for Firmware Release 1.32 Build 0 on page 26.
2. Mandatory: If flashing completes, cold boot the system to refresh all components.

At this point, the controller would be updated and would be ready to use. Install the SmartPQI driver and the latest version of the Arcconf/maxView management utility to monitor and configure the controller.

Note: Downgrading firmware could lead to unexpected behavior due to an incompatibility in SEEPROMs between this release and the prior release.
4 Installing the Drivers

See the "Microsemi Adaptec® HBA 1100 Series Host Bus Adapters Installation and User’s Guide (ESC-2161232)" for complete driver installation instructions.
Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.