Release Notes

SmartHBA 2100 and SmartRAID 3100 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>16</td>
<td>June 2018</td>
<td>SR2.3 Production Release</td>
</tr>
<tr>
<td>15</td>
<td>June 2018</td>
<td>Updated for RC Release</td>
</tr>
<tr>
<td>14</td>
<td>October 2017</td>
<td>Update supported OSs</td>
</tr>
<tr>
<td>13</td>
<td>October 13, 2017</td>
<td>First Production Release</td>
</tr>
<tr>
<td>1-12</td>
<td>June 2016-July 2017</td>
<td>Pre-Production Releases.</td>
</tr>
</tbody>
</table>
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software License Agreement</td>
<td>5</td>
</tr>
<tr>
<td>1 About This Release</td>
<td>8</td>
</tr>
<tr>
<td>1.1 Release Identification</td>
<td>8</td>
</tr>
<tr>
<td>1.2 Components and Documents Included in this Release</td>
<td>8</td>
</tr>
<tr>
<td>1.3 Files Included in this Release</td>
<td>9</td>
</tr>
<tr>
<td>2 What is New?</td>
<td>12</td>
</tr>
<tr>
<td>2.1 Features</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Fixes</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1 Firmware Fixes</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1.1 Fixes and Enhancements for Firmware Release 1.60 B0</td>
<td>13</td>
</tr>
<tr>
<td>2.2.2 UEFI Fixes</td>
<td>23</td>
</tr>
<tr>
<td>2.2.2.1 Fixes and Enhancements for UEFI Build 1.3.4.11/Legacy BIOS Build 1.3.4.6</td>
<td>24</td>
</tr>
<tr>
<td>2.2.3 Driver Fixes</td>
<td>29</td>
</tr>
<tr>
<td>2.2.3.1 Fixes and Enhancements for Linux Driver Build 1.1.4-132</td>
<td>29</td>
</tr>
<tr>
<td>2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 1.0.2-1038</td>
<td>30</td>
</tr>
<tr>
<td>2.2.3.3 Fixes and Enhancements for Solaris Driver Build 1.0.2-1038</td>
<td>31</td>
</tr>
<tr>
<td>2.2.3.4 Fixes and Enhancements for Windows Builds 100.64.2.64 and 6.64.2.64 Pass 1</td>
<td>31</td>
</tr>
<tr>
<td>2.2.3.5 Fixes and Enhancements for VMware Driver Build 1.0.2-1038</td>
<td>31</td>
</tr>
<tr>
<td>2.2.4 Management Software Fixes</td>
<td>32</td>
</tr>
<tr>
<td>2.2.4.1 Fixes and Enhancements for Arcconf/maxView Build B23167</td>
<td>32</td>
</tr>
<tr>
<td>2.3 Limitations</td>
<td>43</td>
</tr>
<tr>
<td>2.3.1 Firmware Limitations</td>
<td>43</td>
</tr>
<tr>
<td>2.3.1.1 Limitations for Firmware Release 1.60 Build 0</td>
<td>43</td>
</tr>
<tr>
<td>2.3.1.2 Limitations for Firmware Release 1.32 Build 0</td>
<td>43</td>
</tr>
<tr>
<td>2.3.2 UEFI Limitations</td>
<td>44</td>
</tr>
<tr>
<td>2.3.2.1 Limitations for UEFI Build 1.3.4.11/Legacy BIOS Build 1.3.4.6</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3 Driver Limitations</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3.1 Limitations for Linux Driver Build 1.1.4-132</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3.2 Limitations for Windows Driver Builds 100.64.2.64 Pass 1 and 6.62.2.64 Pass 1</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3.3 Limitations for FreeBSD Driver Build 1.0.2-1038</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3.4 Limitations for Solaris Driver Build 1.0.2-1038</td>
<td>44</td>
</tr>
<tr>
<td>2.3.3.5 Limitations for VMware Driver 1.0.2-1038</td>
<td>44</td>
</tr>
<tr>
<td>2.3.4 Hardware Limitations</td>
<td>44</td>
</tr>
<tr>
<td>2.3.5 Management Software Limitations</td>
<td>45</td>
</tr>
<tr>
<td>2.3.5.1 Limitations for Arcconf and maxView Build B23167</td>
<td>45</td>
</tr>
<tr>
<td>3 Updating the Board Firmware for PQI Operation</td>
<td>46</td>
</tr>
<tr>
<td>3.1 Updating Controllers to latest (PQI) Firmware</td>
<td>46</td>
</tr>
<tr>
<td>4 Installing the Drivers</td>
<td>47</td>
</tr>
</tbody>
</table>
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 SmartHBA 2100/SmartRAID 3100 controller solutions from Microsemi.

1.1  Release Identification

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

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></td>
<td>RIS version 280</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>
<tr>
<td>arcconf/Maxview</td>
<td>B23167</td>
</tr>
</tbody>
</table>

1.2  Components and Documents Included in this Release

Download the firmware, drivers, host management software, and supporting documentation for your SmartHBA 2100/SmartRAID 3100 controller and SmartRAID 3100 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>Arconfermupdate</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</td>
<td>SmartPqi.sys</td>
<td>x64</td>
</tr>
<tr>
<td></td>
<td>Server 2012, R2 and Windows 8.1, 8</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>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux 7.5¹, 7.4¹, 7.3, 7.2, 7.1, 7.0</td>
<td>x64</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 6.9¹, 6.8, 6.7, 6.6</td>
<td>x64</td>
</tr>
</tbody>
</table>
### Drivers

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuSE Linux Enterprise Server 12&lt;sup&gt;2&lt;/sup&gt;, SP3, SP2, SP1, and Base</td>
<td>x64</td>
</tr>
<tr>
<td>SuSE Linux Enterprise Server 11, SP4 and SP3&lt;sup&gt;1&lt;/sup&gt;</td>
<td>x64</td>
</tr>
<tr>
<td>SuSE Linux Enterprise Server 15&lt;sup&gt;1&lt;/sup&gt;</td>
<td>x64</td>
</tr>
<tr>
<td>Oracle Linux 7.4, 7.3, 7.2</td>
<td>x64</td>
</tr>
<tr>
<td>Oracle Linux 7.3 with UEK 4.1.12-61.1.18</td>
<td>x64</td>
</tr>
<tr>
<td>Oracle Linux 7.2 with UEK 3.10.0-327.el7</td>
<td>x64</td>
</tr>
<tr>
<td>Ubuntu 16.04.3, 16.04.2, 16.04.1, 16.04.0</td>
<td>x64</td>
</tr>
<tr>
<td>Ubuntu 14.04.5, 14.04.4</td>
<td>x64</td>
</tr>
<tr>
<td>Debian 9.0</td>
<td>x64</td>
</tr>
<tr>
<td>Debian 8.8</td>
<td>x64</td>
</tr>
<tr>
<td>Citrix xenServer 7.1</td>
<td>x64</td>
</tr>
</tbody>
</table>

**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".

### 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 Linux x64 VMware EXSi 5.5/6.0 XenServer6.5.1</td>
<td>See the Arcconf download package for the OS-applicable installation executable.</td>
</tr>
<tr>
<td>ARCCONF for UEFI</td>
<td>Windows x64 Linux x64 VMware EXSi 5.5/6.0 XenServer6.5.1</td>
<td>See the firmware download package applicable for the controller.</td>
</tr>
<tr>
<td>maxView Storage Manager</td>
<td>Windows x64 Linux x64 VMware EXSi 5.5/6.0 XenServer6.5.1</td>
<td>See the maxView Storage Manager download package for the OS-applicable installation executable.</td>
</tr>
</tbody>
</table>
### Executable OS Description

See the VMware maxView Storage Manager download package for the OS-applicable installation executable.

### vCenter 5.5 and 6.0 maxView vSphere Plugin

<table>
<thead>
<tr>
<th>Description</th>
<th>OS</th>
<th>Executable</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxView vSphere Plugin</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>Boot USB (offline or pre-boot) for ARCCONF and maxView Storage Manager</td>
<td>Linux x64</td>
<td>See the maxView BootUSB download package for the .iso file.</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>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>Arcconf romupdate</td>
<td>X</td>
</tr>
<tr>
<td>Configurable Big Block Cache Bypass</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>Arccconf Tool Support</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>maxView Tool Support</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Green Backup Support for SmartRAID</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4Kn Support in RAID</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Notes relating to the preceding table: Feature Summary

1. Only applicable for encryption-enabled products.
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 a data corruption when RAID-6 rebuild or parity consistency checking encounters an error that requires a regenerative operation to overcome.
  - **Root Cause**: The state information for the relevant stripe associated with the operation was not being tracked during 'cascaded' operations which only occur when there is no host I/O. These cascaded operations handle the stripe locks differently than normal host operations, but that stripe information is needed if the cascade operation subsequently requires further error handling to be able to complete. In this case, the stripe information being stale would lead to the regenerative logic incorrectly pulling data from elsewhere in the RAID set.
  - **Fix**: Fill in the stripe information during cascade operations.
  - **Risk**: Low

- 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 an issue where fault LED for a disk is not lit upon boot up discovery.
  - **Root Cause**: State variables governing when it's appropriate to indicate fault were not set prior to initial discovery, but were being set prior to hotplug discovery. This inconsistency cause the fault LED to behave differently between the two disk discovery scenarios.
  - **Fix**: Modified the boot up discovery state variables initialization to be the same as the hotplug code path.
  - **Risk**: Medium

- Fixed a potential controller lockup when firmware completes Rapid Parity Initialization of a newly-configured RAID volume.
  - **Root Cause**: A race condition was identified where two threads were both attempting to notify the host with an event which caused the same host notification resource to be used twice.
  - **Fix**: Improved the locking on the event queue such that only one event notification is able to be processed at a time.
  - **Risk**: Low
- Fixed an issue where the controller takes a long time to delete volumes while also performing rebuild and encountering disk errors.
  - **Root Cause:** If the rebuild state machine has reverted to single block regeneration due to encountering disk errors, it could take a long time to exit that main loop before noticing that this operation should be stopped to continue processing the user configuration change request.
  - **Fix:** Each time through this single-block regenerative loop, check to see if there is a pending request that needs to be serviced. Firmware can yield here and resume later if this volume isn’t destroyed as a result of the pending configuration change.
  - **Risk:** Low

- Fixed a controller lockup 0xFFFFF001 while removing and adding drives in a RAID-5 or RAID-50 set while rebuild is occurring.
  - **Root Cause:** A race condition was leading to a rebuild request incorrectly referencing disk parity requests built for targets which has since been removed, resulting in firmware lockup.
  - **Fix:** Modified the parity rebuild logic to abandon submitting the request so that the downstream logic won’t act on the missing information.
  - **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 controller lockup during a device or I/O failure with a 4Kn RAID volume configured.
  - **Root Cause:** In cases where a host request had been promoted into a read-ahead for the DDR caching layer, the block count of the original request would be modified. In this case, one such request was ongoing at the time of the failure or URE response of a drive. Error handling for this scenario is to abandon the read-ahead and retry the original host operation, but in this case the block conversion back to the original size was not occurring. This leads to a hardware exception when the size of the request was compared against the buffer provided.
  - **Fix:** Properly revert the block count in the error handler before attempting the retry.
  - **Risk:** Medium

- 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 the host is unable to control LED states for expander-attached HBA targets.
  - **Root Cause:** Because both the host and firmware may be simultaneously managing SES content for a device, the firmware attempts to collect and maintain a unified set of content for sending to the target. In this case, firmware was overwriting the host-provided SES display element data with data of its own, even when that display element is associated with a drive not in a RAID set.
  - **Fix:** Modified the SES content merge logic to only allow firmware to overwrite the data for elements associated with RAID sets.
  - **Risk:** Low
• Modified the "known bad blocks" tracking for RAID volume members to track explicit blocks rather than block ranges. This will clean up debug logs and simplifies some of the logic associated with tracking encountered or marked URE's as well as the logic employed to avoid these blocks.

• Fixed a potential for a volume to be taken offline by Linux kernels after a LUN reset.
  - **Root Cause:** In a configuration with multiple volumes having DDR caching enabled and the cache being full of data, the firmware will prefer to continue providing free lines as they become available to the more active volume. In a case where one volume is very active and a second is sparsely active, it is possible for the sparsely active volume to have significant waits before its I/O is serviced. If the host sends a LUN reset for this volume but also continues to send IOs to the busier volume, it's also possible for the LUN reset to take longer because any previous requests still waiting for room in the cache must be drained before the LUN reset can complete. Once that operation completes, the original state occurs again and some Linux kernels will decide to take the volume offline as a quality-of-service choice.
  - **Fix:** Instead of waiting forever for room in the cache, time waiting requests and force each to bypass the cache if it waits longer than 5s.
  - **Risk:** Medium

• 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 an invalid replacement drive for a degraded RAID set would not be made available for configuration if that RAID set was subsequently destroyed.
  - **Root Cause:** A drive self-test kicked off when an invalid replacement scenario was encountered was still ongoing at the time the RAID volume was being deleted. This prevented actions necessary to bring that drive back online from occurring successfully, leading to the drive still being left in a failed state.
  - **Fix:** A drive self-test is not necessary for this scenario because there is no fault with the drive, so this procedure was removed for configuration related drive offline scenarios.
  - **Risk:** Low

• Fixed an issue where host request coalescing was not being enabled for non-parity RAID volume types when DDR caching is disabled.
  - **Root Cause:** A fix for a coalescing issue also attempted to simplify the ruleset employed to govern when firmware should or should not be allowing host request coalescing to take place. This change also modified the behavior such that these volume types were disqualified from coalescing, though that was not clear based on the original code either.
  - **Fix:** Allow coalescing for all volume types when DDR cache is disabled. Additional detail of the volume coalescing state was added to various debug outputs to make quick identification of this performance functionality more obvious during support dump analysis.
  - **Risk:** Medium

• Fixed an issue where the fault LED could be incorrectly set on a drive if a RAID volume was undergoing transformation and drives were being removed and re-inserted from both the old and new RAID set configurations.
• **Root Cause:** Firmware was not properly updating the "old" RAID set configuration during the hot-add processing for a drive which subsequently left it's position in the set as being in a fault status.

• **Fix:** Ensure both the old and new RAID set configurations are updated when volume state changes are occurring as a result of drive hot-add processing.

• **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 lockup 0x1C50 when multiple rebuild IO's are outstanding to drives in a JBOD which is spuriously power cycled.

• **Root Cause:** The rebuild state machine did not have handling for cases where both read and write IO's could fail at the same time.

• **Fix:** The write errors on their own will result in target drive failure, so added logic to ignore subsequent read errors when write errors are occurring. In this case, the drive would also be failed due to hot-removal but the rebuild state machine is encountering this state first in response to some of the IO clean-up occurring.

• **Risk:** Low

Fixed an issue where the second drive replacement added to a double-degraded RAID-6 volume would not result in that drive's LED state blinking to indicate rebuild in progress.

• **Root Cause:** The LED state info was being set during the drive hot-add, but was not updated at the start of the rebuild state machine. This resulted in the first replacement drive LED operating as expected but the second drive replacement not having the correct behavior.

• **Fix:** Added logic to update the LED states at the start of the rebuild state machine.

• **Risk:** Low

• Fixed an issue where a newly created RAID volume undergoing rapid parity Initialization would still be exposed to the host OS.

• **Root Cause:** The code which presents the host presentation status of a RAID volume was not taking its initialization state into account. This would lead to undesirable behaviors when the host tried to access the volume, for example, repeated LUN resets.

• **Fix:** Added appropriate logic to suppress the RAID volumes in a not-accessible state from the host OS presentation, and then transition them to available once the parity initialization process is completed.

• **Risk:** Low

• Fixed an issue where data and spare drive identities were being mixed up when a RAID volume from another controller is being merged into an existing RAID set, while also moving drive members of that existing RAID set to new positions.

• **Root Cause:** The newly migrated drive RAID metadata signature was being invalidated because it did not match the previously native RAID set. The subsequent handling of the movement of the native set became confusing as the drives were still present in the old positions.

• **Fix:** Separated the logic to validate discovered configuration data, such that both the native set movement handling and the import of the new metadata are allowed to complete successfully.
• Risk: Medium

• Fixed an issue where drives in a dual domain JBOD appear twice for user selection in RAID volume creation after power cycling the JBOD.
  • Root Cause: The JBOD does not appropriately report correct enclosure logical identifiers that causes firmware to disassociate the expanders in each domain with each other and thus the drive paths not with each other. The end result is that the each path to the drives appears as different drives in the user presentation.
  • Fix: A workaround was to reconcile the two expanders for discovery during boot, but the required hints were not available during hot-plug in time to cause the path information to be coalesced correctly. For hotplug, we added WWN of the expander to the 'failed expanders' bookkeeping such that on hot-add we have the additional information we need to know that this configuration is actually a single JBOD rather than two disparate ones.
  • Risk: Medium

• Fixed an issue where the green backup recovery is allowed to be attempted even though the green backup state is not dirty.
  • Root Cause: The green backup layer fails attempts to clean up when no clean up is necessary, which generates unnecessary error outputs and status to be reported.
  • Fix: Do not allow this recovery process when it is not necessary.
  • Risk: Low

• Fixed an issue where a green backup hardware error is prematurely reported after a real backup error has occurred.
  • Root Cause: When the backup system is in a "bricked" state due to a backup error, the green backup API initialization call returns a failure. This is the correct behavior, but the fault status was being saved without knowing the type of the error.
  • Fix: Inspect the hardware states and address any fixable conditions before reporting the error, so that more accurate reasoning associated with the error state can be provided.
  • 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 controller UART log was spammed with the drive failure messages after a drive is hot-removed from a parity RAID volume.
  • Root Cause: A path identified in the IO recovery path for parity volumes was not taking into account that the state of the disk may have changed between mapping the request and attempting to perform the disk operations. This lead to error handling deciding the (now missing) disk should be marked failed (again) for each of the IO's that were falling into recovery as a part of the original disk failure.
  • Fix: Added the appropriate checks to this IO path, such that it can bail out and perform more appropriate recovery earlier based on the knowledge that a related drive is no longer present.
  • Risk: Low
• Fixed an issue where a legacy drive erase process was abandoned due to the drive hot-removal could be re-started on the next controller boot even if that drive had been replaced.
  • Root Cause: The state information to indicate the erase being aborted was not being stored in the RAID metadata promptly. This would lead the controller to believe that the erase needed to be resumed during the next boot.
  • Fix: Explicitly save the RAID metadata when this process is aborted.
  • 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 a system appears to hang while repeatedly encountering URE errors from disks.
  • Root Cause: Controller error recovery was attempting to recover data, but the slowness of the drive was being magnified by encountering many UREs.
  • Fix: On reads, bail out earlier.
  • Risk: Low

• Fixed a controller hang if the host submits a read to a logical volume with zero for block count requested.
  • Root Cause: Firmware was calculating the end block of the request which would result in an underflow value prior to entering a loop to start generating requests for the disks. This underflow caused the loop to execute a huge number of iterations, effectively causing the controller to appear to be hung.
  • Fix: The T10 standard indicates a requested count of zero should result in the operation simply returning SUCCESS status, so the IO handling was updated to match this requirement.
  • Risk: Low

• Fixed a controller hang for the MCM-based designs where the RAID volumes from other controllers are being merged into the current configuration during boot.
  • Root Cause: The memory pool used for this operation was not large enough to accommodate the size of the buffer required to merge several different configurations at once. The code would hang waiting on the allocation that could never succeed as the buffer pool could not fulfil the request.
  • Fix: Moved the allocation to a different, much larger, memory pool.
  • Risk: Medium

• 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 several controller lockup 0x1E30 issues associated with enabling DDR caching while performing a sequential IO workload to a RAID volume.
  • Root Cause: The logic to determine the time to coalesce the host requests was looking at both the global DDR cache state as well as the per-volume DDR cache policy. When DDR cache is temporarily disabled, for example, while the backup power source is charging, this check results in coalescing as ok to perform. If the DDR cache is suddenly re-enabled while a coalesced request has not yet passed the caching layer in the IO path, firmware would lock up.
  • Fix: In addition to simplifying and optimizing the checks related to the way the DDR caching relates to the host request coalescing to help with performance, also allows coalescing to not be enabled for a volume which is allowed to use DDR cache even though it is temporarily disabled as these are typically transient states lasting only a few seconds or minutes, for example, waiting for the power source to charge.
  • Risk: Low

• Fixed an issue where the host request coalescing was incorrectly disabled for all the volumes whenever there is a maxCache volume configured.
  • Root Cause: The maxCache creation code was setting a controller-level policy to disable the host request coalescing, and that policy would be cleared when the last maxCache volume is deleted. There was no reason to be so restrictive with this policy. However, as it arbitrarily penalizes other volumes that could be using this IO optimization.
  • Fix: Removed the logic from the maxCache creation code to set this controller policy/ Since the I/O path already makes the correct determinations regarding when it is appropriate to allow a volume to coalesce host requests.
  • Risk: Low

• Fixed an issue where configuring unassigned drives into new RAID sets could become blocked on a mixed mode port configuration if there are failed RAID sets.
  • Root Cause: Firmware was incorrectly reporting the drive connector configuration status as UNKNOWN for the failed drives. The host tools would subsequently disable the configuration of drives on that connector because the ruleset for the configuration was being presented inconsistently.
  • Fix: Report the configured controller connector mode regardless of the status of the drive, since that status is not relevant to the configuration rules related to other non-failed drives.
  • 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 an issue where an HBA disk could be spuriously re-assigned as a RAID volume spare after two controller reboots.
  • Root Cause: Legacy behavior for RAID-only mode connectors would pick up available unconfigured disks as spare replacements if a spare drive became spuriously missing after a reboot. It takes two reboots because the volume would be flagged as LOOSE_CABLE in the
first boot that the UEFI driver would acknowledge, after which the spare assignment would occur on the next boot.

- **Fix:** Although firmware was already tracking HBA disk assignments, some state information was not populated because some disks were HBA targets rather than RAID-only unconfigured disks. The fix was to both ensure this state info is populated as well as to restrict automatic assignment only to the unconfigured RAID devices and not use any drive, currently exposed to the host as an HBA target.
  - **Risk:** Low

- **Fix:** An issue where the controller firmware freezes waiting for a response from the host to a Vendor Defined Message (VDM) request.
  - **Root Cause:** An outbound (controller to host) VDM message can get stuck in a controller queue and not sent to the PCIe bus because interrupts for that queue became permanently disabled when two threads simultaneously accessed the interrupt register. Each thread's access profile to the interrupt register was a non-atomic read-modify-write.
  - **Fix:** Modified controller firmware implementation so each firmware thread's access to the interrupt register is atomic.
  - **Exposure:** Only on systems that uses VDM.
  - **Risk:** None

- **Fix:** 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

- **Fix:** 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 problem where a transforming volume fails or never resumes transformation after graceful system shutdown and reboot.
  • **Root Cause:** The backup power subsystem was not being re-armed after shutdown. Previous transformation models forced in-flight shutdowns to be dirty, however the process was modified to save transformation progress to disk to allow graceful shutdown.
  • **Fix:** Added the missing logic required to re-arm the backup power subsystem and resume volume transformation after powering on after graceful shutdown.
  • **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 the VPD page 83h data is not properly reset after a "heal array" operation is performed on a volume marked as a primary/secondary boot target.
  • **Root Cause:** In handling the RAID metadata update, the primary and secondary boot ID’s were being cleared.
  • **Fix:** Preserve the primary and secondary boot volume identifiers such that they can be re-used in creation of the volume's VPD page 83h data, if needed.
  • **Risk:** Low

• 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 the potential incorrect supercap connection status if the cache backup firmware has failed.
  
  **Root Cause:** Detaching supercap during a Backup will cause the green backup subsystem in firmware to fail and not startup on the subsequent server boot; as a result, the firmware will not update the supercap status information. Therefore, a query by host management software will incorrectly return supercap connection status as "connected".

  **Fix:** When host management software queries the cache backup system status, the firmware will read a general purpose IO pin to get the supercap connection status to update the connection status and return it to host management software tools; as result, the tools will get the most up-to-date supercap connection status.

• **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).

• Increased available MaxCache LUN size.

• 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 legacy BIOS fixes and enhancements:

- Fixed an issue where a Rapid Parity Initialization task is not running for RAID 50 and RAID 60.
  - **Root Cause:** The RAID level comparison for setting the RPI bit failed.
  - **Fix:** Updated the condition check for the RAID level.
  - **Exposure:** All previous versions.
  - **Risk:** Low.

- Fixed an issue where the backup power status indicates "Charged" state in CTRL+A utility while its actual state is charging.
  - **Root Cause:** The backup power status check, used to display the charged status, was always returning true.
  - **Fix:** Corrected status check logic.
  - **Exposure:** All previous versions.
  - **Risk:** Low.

- Added controller-based encryption support for legacy BIOS Ctrl-A.
  - Fixed an issue to create a logical drive under legacy mode, if change raid level before press "Done", then the created logical level always be RAID 0.
    - **Root Cause:** A condition check before assigning the value to the response field was missing and response was getting set to RAID 0.
    - **Fix:** Added appropriate check.
    - **Exposure:** All previous versions.
    - **Risk:** Low.

This release includes the following UEFI fixes and enhancements:

- Fixed an issue where user is unable to configure MaxCache of capacity greater than 1.8 TB using UEFI HII.
  - **Root Cause:** Improper order for setting cache line size and maxCache size.
  - **Fix:** Cache line size set prior to setting maxCache size.
  - **Exposure:** All previous versions.
  - **Risk:** Low.

- 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 a failure to create 17 drive RAID 5 maxCache logical drive using 64 KiB CLS.
  - **Root Cause:** Unless client/user explicitly uses SA_SetEditableLogicalDriveLUCacheLineSize(), storageCore defaults to using 64 KiB. However if an maxCache logical drive is configured with
RAID 5 on an Array with more than 9 drives (17 or 33), 64 KiB is not a valid cache line size and the firmware command will fail to create the maxCache logical drive. Additionally, SA_CanSetEditableLogicalDriveUICacheLineSize() is allow users to set cache line sizes that not supported by the size of the Array and RAID level such as setting 64 KiB for a 17 PD Array with RAID 5.

- **Fix:** SA_SetEditableLogicalDriveRAIDLevel() will properly set the cache line size to 256 KiB if 64 KiB is not valid for the configuration. Also add a check to SA_SetEditableLogicalDriveUICacheLineSize() to check the Array size and RAID level. If the desired cache line size cannot be set for this check the API will return a new error.

- **Exposure:** All previous versions.
- **Risk:** Low.

- Fixed an issue where users are allowed to set the logical drive label to an empty value.
  - **Root Cause:** SA_CanSetEditableLogicalDriveLabel() only checks the buffer size of the input but does not check the length of the desired label.
  - **Fix:** Add an additional check for the length of the desired label in SA_CanSetEditableLogicalDriveLabel(). If the length is 0 return kCanSetEditableLogicalDriveLabelInvalidLabel.

- **Exposure:** All previous versions.
- **Risk:** Low.

- Fixed an issue where a logical drive status strings does not match with strings of other host tools.
  - **Root Cause:** Logical drive status strings are different compared to strings of other host tools.
  - **Fix:** Logical drive status strings made same as strings of other host tools.

- **Exposure:** All previous versions.
- **Risk:** Low.

- Fixed an issue where maxCache creation is not possible when controller is configured with 12 drive 3 parity group RAID 50.
  - **Root Cause:** ParityCount was having stale data and not initialized for maxCache logical drive creation.
  - **Fix:** ParityCount initialized to 0 in RAID selection page.

- **Exposure:** All previous versions.
- **Risk:** Low.

- Fixed an issue where create a logical drive using SAS drives with redundant path fails.
  - **Root Cause:** If the POST request uses a dual-path notation, software finds a drive with a given path that matches one of the values in the POST request. However, when drives are zoned to a controller that supports long connector names, the LOCATION of the drive does not match any of the report PATHs.
  - **Fix:** If the POST request uses a dual-path notation, check all given values with all PATHs of a drive and the LOCATION of a drive.

- **Exposure:** All previous versions.
- **Risk:** Low.

- Fixed an issue where if the controller cache is temporarily or permanently disabled, the MaxCache creation is not allowed and an error message is reported.
  - **Root Cause:** A check for controller cache status is not bad at the onset of MaxCache creation.
  - **Fix:** Controller cache status is now checked at the beginning of MaxCache creation. If the controller cache is temporarily or permanently disabled, the MaxCache creation is not allowed and an error message is displayed.

- **Exposure:** All previous releases.
• Risk: Low

• Fixed issues in modifying cache ratio value which is not multiple of 5.
  • Root Cause: Function call responsible for checking enablement of write and read cache ratio weren’t correct.
  • Fix: Modified the responsible function to implement proper checking of cache ratio.
  • Exposure: All previous versions
  • Risk: Low

• Fixed an issue where the selected acceleration method ‘none’ is not applied during logical creation.
  • Root Cause: When the battery is absent, the write cache size is greater than zero, and No Battery Write Cache mode is turned off, the controller does not properly set the acceleration method.
  • Fix: Set the write cache size to zero, if:
    • The battery is absent.
    • The write cache size is greater than zero.
    • No Battery Write Cache mode is turned off.
  • Exposure: All previous versions
  • Risk: Medium

• Fixed an issue where there is no option for cache_line_size selection during MaxCache creation.
  • Root Cause: Cache_line_size selection option is not provided for MaxCache creation.
  • Fix: Cache_line_size selection option is provided for MaxCache creation.
  • Exposure: All previous releases
  • Risk: Low

• Fixed an issue where the logical drives are not listed when creating RAID 5 MaxCache logical drive with the write policy as write through.
  • Root Cause: CacheRAIDPolicy support is not considered for cache policy menu construction.
  • Fix: CacheRAIDPolicy support is now considered for cache policy menu construction.
  • 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

• Fixed an issue in Array I/O accelerator function when RPI in progress.
  • Root Cause: There was no check in the function responsible for setting Array IOBypass for logical drives when they are undergoing RPI.
  • Fix: In Array IOBypass function added a check for any logical drives undergoing RPI. If so, return an error.
  • 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 where HII has to show proper error message for MaxCache creation failure when controller cache is temporarily or permanently disabled.
  • Root Cause: A check for controller cache status is not bad at the onset of MaxCache creation.
  • Fix: Controller cache status is now checked at the beginning of MaxCache creation. If the controller cache is temporarily or permanently disabled the MaxCache creation is not allowed with error message given.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed issues in modifying cache ratio value which is not multiple of 5.
  • Root Cause: Function call responsible for checking enablement of write and read cache ratio weren't correct.
  • Fix: Modified the responsible function to implement proper checking of cache ratio.
  • Exposure: All previous versions
  • Risk: Low

• Fixed an issue where the selected acceleration method 'none' is not applied during logical creation.
  • Root Cause: When no battery is present, the write cache size is greater than zero, and No Battery Write Cache is turned off the controller would not properly set the acceleration method.
  • Fix: Set the write cache size to 0 if:
    • The battery is not present.
    • The write cache size is greater than zero.
    • No Battery Write Cache is turned off.
  • Exposure: All previous versions
  • Risk: Medium

• Fixed an issue where improper text was displayed for driver health message 1786.
  • Root Cause: 1786 driver health message "Logical drive(s) previously failed" is wrong.
  • Fix: 1786 driver health message changed to "Recovery Needed".
  • Exposure: All previous releases
  • Risk: Low

• Fixed an issue where there is no option for cache_line_size selection during MaxCache creation.
  • Root Cause: Cache_line_size selection option not provided for MaxCache creation.
  • Fix: Cache_line_size selection option provided for SmartCache creation.
  • Exposure: All previous releases
  • 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 Array I/O accelerator modifiable when RPI in progress.
  • Root Cause: There was no check in the function responsible for setting Array IOBypass for logical drives when they are undergoing RPI.
  • Fix: Added check in Array IOBypass function to check if any logical drives are undergoing RPI and if so return new error: kCanSetEditableArrayIOBypassHasInitializingLogicalDrive.
  • Exposure: All previous versions
  • Risk: Low

• Added an enhancement to provide MaxCache support in UEFI driver.
  • Implementation: HII menu and options added to create, delete, and edit MaxCache arrays.
  • Risk: Medium.

• Fixed an issue where RIS is failing to change the controller drive write cache configuration.
  • Root Cause: In Permissive mode, logical drives can exist on the controller without specifying any logical drives in the LogicalDrive array. This would cause modifications to any controller setting to fail because the settings are expected to be 'null'.
  • Fix: When DataGuard is set to Permissive, allow non-null values for controller settings, even if the LogicalDrives array is empty. Note that if no logical drives are present at all on the controller, the request will still correctly fail.
  • Exposure: All previous versions.
  • Risk: Low.

• Fixed an issue to create a RAID 0 logical drive with a single spare while setting the spare activation mode to "Predictive" may fail to create the logical drive and report no errors.
  • Root Cause: If no logical drive has ever been configured since the previous boot, the controller cannot save the predictive spare activation setting prior to creating any new logical drives. The RAID 0 logical drive with spare command is rejected by firmware in this scenario.
  • Fix: Change the request process sequence such that when the command to create a RAID 0 logical drive with spares fails and no logical drives existed when the editable configuration was initialized, re-attempt to create the logical drive without spares and queue spare management until later during the commit (after the predictive spare activation is set).
  • Exposure: All previous versions.
  • Risk: Low.

• Fixed an issue to differentiate between the logical array and MaxCache array in array configuration menu.
  • Root Cause: There is no way to differentiate array type in array list menu.
  • Fix: Array type has been added to the array list string.
  • 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.

• Fixed an issue where the Offset Logical drive creation is not successful while created using RPI method.
  • Root Cause: BuildMethod is not getting reinitialised for new array creation.
  • Fix: BuildMethod is initialised to default and updated to the value as set during array creation.
  • Exposure: All previous versions.
  • Risk: Low.

• Added controller based encryption support for UEFI HII.

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.
• Fixed an issue with handling of synchronous requests.
  • Root Cause:
    1. The driver was not detecting some rare error cases for synchronous requests down the RAID path.
    2. The driver was not retrying the INQUIRY of VPD page 0 sent to an HBA drive, if the command failed due to an abort.
  • Fix: Improved error checking for synchronous requests and added retries for INQUIRY of VPD page 0.
  • Risk: Low
  • Exposure: The driver behavior has been the same since the out-of-box version of SmartPQI was released. The probability of this issue occurring is very low because it requires a number
of factors (a SATA drive with outstanding NCQ commands queued to it fails (the drive stops responding) and at the same time an INQUIRY command is submitted to the drive) to align perfectly to cause the conditions that expose the issue.

- 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 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 SCSI status 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.
  • **Risk**: Low

### 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 an issue where the VMware driver caused a PSOD due to DE exception.
• **Root Cause:** Stripe size or disks per row given by the firmware raid map is zero. This value is used as divisor when calculating the raid bypass parameters.
• **Fix:** Added condition to check for zero in bypass RAID IO path at all possible places. If condition is met, it will send the IO in RAID path itself.

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 IOPV 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 Arconf/maxView Build B23167
This release includes the following fixes for Arconf/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 issue to allow creation of cache size based on system configurations and options presented to user.
  • **Root cause:** A new field for Cache Line Size has been added during the creation of maxCache.
  • **Fix:** Made proper changes to take care of cache Line Size during the creation of maxCache.
  • **Exposure:** All previous releases.
  • **Risk:** Low
• Fixed an arcconf 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 logical device status change for RAID10 devices.
  • Root cause: Need to change display information on RAID10 and RAID 10ADM.
  • Fix: Modified RAID 10 and RAID10ADM information to display the segment information.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue where the "consistency check inconsistency notification" property was not updated in GUI.
  • Root cause: Event was missing for "consistency check inconsistency notification".
  • Fix: Added event for "consistency check inconsistency notification".
  • 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 in which the arcconf/maxView switched the controller mode to the RAID mode when OS is installed on the RAW device.
  • Root cause: When OS is installed on device, arcconf switches the controller mode to RAID.
  • Fix: Blocked the switching of the controller mode and the connector mode to RAID when there is partition on device.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue where during the maxCache creation the user was able to select multiple logical device.
  • Root cause: Multiple selection of logical devices were allowed during the maxCache creation.
  • Fix: Modified to limit user to select only one logical device during the maxCache creation.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which arcconf failed to create logical device when there were more than 60 physical devices.
What is New?

• **Root cause**: When the stripe size was not explicitly specified, the default stripe size was set, but the create command was aborted in the validator layer while checking for the possible stripe size.
• **Fix**: When user doesn't specify stripe size explicitly, stripe size is set as maximum possible stripe size.
• **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 array RAID10 or RAID10(ADM) could not get the array hot spare information by arcconf.
  • **Root cause**: `getconfig 1 ld` was not showing the hot spare information for the logical device with RAID 10, 50, and 60.
  • **Fix**: Added code for showing the hot spare information along with the logical device in the `getconfig 1 ld` information.
  • **Exposure**: All previous releases.
  • **Risk**: Low

• Fixed an issue where maxView allowed the split mirror operation when the array had RAID10(ADM) and RAID10.
  • **Root cause**: The check for disabling the `splitMirrorBackup` operation when array has mixed logical devices was missing.
  • **Fix**: Added a check to block `splitMirrorBackup` operation when array has mixed logical devices.
  • **Exposure**: All previous releases.
  • **Risk**: Low

• Fixed an issue where arcconf allows to move a 4k array to 512 block size array.
  • **Root cause**: The move array operation from the 4k array to the 512 block size array failed and gave general failure message instead of a specific error message.
  • **Fix**: Added check for blocking the move array operation with different block size array.
  • **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 arcconf displays a wrong message "Reboot required" for controller mode change operation.
  - **Root cause**: The check for the runtime connector mode changeable feature was missing.
  - **Fix**: Added the proper check for the runtime connector mode changeable feature before showing the required reboot message.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue in which the modify controller cache function for split mirror primary was blocked.
  - **Root cause**: Modifying the controller cache function on the split mirror primary logical device was not allowed.
  - **Fix**: Changes are made to allow the user to modify the controller caching on the split mirror set primary logical device and display a warning message when the user tries to modify the controller caching.
  - **Exposure**: All previous releases.
  - **Risk**: Low

- Fixed an issue in which the arcconf did not allow to set the cache setting on the logical device created with SSDs.
  - **Root cause**: The cache setting on the logical device was blocked if the array interface type was SAS-SSD/SATA-SSD.
  - **Fix**: Removed the wrong check if interface array type is SAS-SSD/SATA-SSD while setting cache property.
  - **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 "command failed" message was displayed for the split mirror operation on an array that had the mixed RAID levels.
  - **Root cause**: Needed "command aborted" message when split mirror operation is performed on an array with mixed RAID levels.
  - **Fix**: Added "command aborted" instead of "command failed" message for display.
  - **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.
What is New?

- **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 there was unsupported error (general failure) displayed while trying to enable the cache setting on the logical device created with SSDs.
- **Root cause:** When SSD IO bypass was enabled on array, call to set the controller caching on the same array was sent instead of being blocked.
- **Fix:** Added check in the validation layer of `arcconf` to display valid error message.
- **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 **Delete logical** icon was disabled for the logical device when the RPI task was ongoing and other logical device has maxCache associated to it.
  - **Root cause:** The condition to handle the deletion of logical device was checking any maxCache on the controller instead of associated one.
  - **Fix:** Made changes to check whether the logical device has associated maxCache.
  - **Exposure:** All previous releases.
  - **Risk:** Low

• Fixed an issue in where there was a button missing in maxView to recover the failed cache module.
  - **Root cause:** Recover cache module feature was not implemented in maxView.
  - **Fix:** Implemented the recover cache module method in STORLIB, and made changes in maxView to display the button for the operation.
  - **Exposure:** All previous releases.
  - **Risk:** Low

• Fixed an issue where the migrate logical device indicated "no record found" for the RAID level selection and the set attributes page.
  - **Root cause:** The check to disable the expand/migrate and to move the logical ribbon icons when the array has a failed logical device(s) was missing.
  - **Fix:** Added check to disable the expand/migrate and move the logical links when the corresponding array has a failed logical device in it.
  - **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 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 where the **Select all** check box was remaining selected when the user was traversing back during maxCache creation.
  • **Root cause**: The value of the check box that has to be enabled/disabled is not set on click of next button.
  • **Fix**: The value of the **Select All** check box is set to false on click of next button to clear it.
• **Exposure**: All previous releases.
• **Risk**: Low

• Fixed an issue where the logical device creation on a new array option was not disabled when the ready devices were not available.
  • **Root cause**: The check to disable the create logical device icon was not available when there were no ready devices available and no free space to create on existing arrays.
  • **Fix**: The check has been added to disable the create logical when there are no ready devices available and no free space to create on existing arrays.
• **Exposure**: All previous releases.
• **Risk**: Low

• Fixed an issue where the **Locate maxCache Help** opened series-8 Locate maxCache help page.
  • **Root cause**: Locate maxCache context ID of the Smart controller was referring to the Series-8 controller.
  • **Fix**: Added the proper contextID for **Locate maxCache Help** to refer the Smart array controller.
• **Exposure**: All previous releases.
• **Risk**: Low

• Fixed an issue in which the maxCache size value was accepting multiple comma separators.
  • **Root cause**: The check for validating the size based on special characters between the numerical was missing.
  • **Fix**: The check is added to validate the size based on special characters and a warning is displayed with the valid size information.
• **Exposure**: All previous releases.
• **Risk**: Low

• Fixed an issue in which the logical device in failed state was listed for selection in the create maxCache logical selection page.
  • **Root cause**: Check was missing for forming the tree in the create maxCache wizard for the failed logical device.
  • **Fix**: Added check while forming the tree table in create maxCache wizard to prevent the listing of the failed logical device.
• **Exposure**: All previous releases.
• **Risk**: Low

• Fixed an issue in which the reboot message was missing in arcconf and maxView, when recovering the failed cache module.
  • **Root cause**: Reboot message was missing along with the confirmation before executing the command. In arcconf, the warning message to inform the user about requiring a reboot for recovering cache module operation was missing in maxView.
  • **Fix**: Added the code to display a reboot message and getting a user confirmation to proceed recovering the cache module. Added a reboot warning message after clicking the **Recover Cache Module** button in the **Set Properties Controller** dialog.
• **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 Spare Management icon was disabled for a predictive failure physical device that already was assigned as dedicated spare to an array.
  • Root cause: The state of the physical device was "Predictive failure" which was skipping the checks resulting in disabling Spare Management icon.
  • Fix: Added check for referring the configuration type along with state of the physical device to enable the Spare Management ribbon icon.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which the Migrate RAID 6 to RAID 50 operation was failing.
  • Root cause: During the migrate operation from RAID 6 to RAID 50, sub-array count was not rendered and default value is set to 1.
  • Fix: Added the proper sub-array count when RAID50 and RAID60 is selected during migration and display the sub-array count in the Attributes page.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which the Information icon on the max logical device size for RAID50/60 creation was always based on the sub-array count 2.
  • Root cause: The value for size in the tooltip of Information icon was not rendered when the sub-array count was changed.
  • Fix: The Information icon was updated once the value of size was changed based on the chosen sub-array count.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which the requirement was to disable logical creation on new array, when the ready devices were not available.
  • Root cause: In maxView, logic to disable the create logical device operation on new array option was missing, in the absence of ready physical devices.
  • Fix: Added logic device to disable create logical device operation on new array option, in the absence of ready physical devices.
  • 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.
  • 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 the **Next** and **Previous** event buttons were grayed out from the **Logical Device** selection.
  • **Root cause**: Check was missing for the **Next** and **Previous** button to enable when the event dialog was displayed.
  • **Fix**: Added the proper check to enable and disable the **Next** and **Previous** button on the event dialog.
  • **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 it was not possible to create maxCache with proposed MAX size in German version of OS.
  • **Root cause**: The group separator as well as the decimal separators were not fetched according to the current locale, and the formatting of the size string was not according to locale specified.
  • **Fix**: Made changes to use group and decimal separators based on locale and to parse the size string according to the locale specified to display the valid MAX size.
  • **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 it was required to modify error message for unsupported devices during RAID 5 maxCache creation.
  • **Root cause**: The response displayed was not generic and does not convey the reason of failure.
  • **Fix**: Changed the to a more generic message as "The RAID configuration specified is not supported on the array".
  • **Exposure**: All previous releases.
  • **Risk**: Low

• Fixed an issue in which the spare management was not listing the predictive failure device while changing from auto replace spare to dedicated spare.
  • **Root cause**: When trying to access the state of predictive failure device, the state was "Predictive failure" so it was not adding to the list as it was not considered as a "Ready" device.
  • **Fix**: Added the proper condition for the predictive failure device by referring to the configuration type to list in the tree.
  • **Exposure**: All previous releases.
  • **Risk**: Low
• Fixed an issue where when maxView reported status as successful, when the controller logged it as failed issuing “recover cache module” for the cache permanent disable condition.
  - **Root cause:** In maxView, the return status to be displayed in the status dialog was not handled properly for recover cache module feature in the set controller properties dialog.
  - **Fix:** Made changes to add the return status of recover cache module feature in the set controller properties dialog.
  - **Exposure:** All previous releases.
  - **Risk:** Low

• Fixed an issue in which the auto-replaceable spare reports failure message, though the spare device was successfully assigned to array.
  - **Root cause:** Assigning a spare and setting the spare type are two different operations in maxView, where the setting spare type to auto-replaceable failed to apply and maxView displayed an error message.
  - **Fix:** Assigning a spare and setting the spare type are implemented as single operation to allow both operations to happen simultaneously.
  - **Exposure:** All previous releases.
  - **Risk:** Low

• Fixed an issue in which the auto-replaceable spare was getting assigned as dedicated spare when PSA was enabled.
  - **Root cause:** Assigning a spare and setting the spare type are two different operations in maxView, where the setting spare type to auto-replaceable failed to apply and maxView displayed an error message.
  - **Fix:** Assigning a spare and setting the spare type are implemented as single operation to allow both operations to happen simultaneously.
  - **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 the functions such as, modify no battery write cache and wait for cache room failed were not supported for this controller.
  - **Root cause:** The **Waitforcacheroom** and **nobotterywritecache** operations are not allowed if all the logical devices are configured with maxCache or "SSD IO Bypass".
  - **Fix:** Implemented a logic to find the number of logical devices configured either with maxCache or SSD I/O bypass to validate the operations support.
  - **Exposure:** All previous releases.
  - **Risk:** Low

• Fixed an issue in which the connector change events were not generated in maxView.
  - **Root cause:** Events for hardrive was not generated as to allow appropriate update of the configuration; sasconnector object was not updated before the harddrive object.
  - **Fix:** Changed the order of updating sasconnector and harddrive objects for generating events as expected.
  - **Exposure:** All previous releases.
  - **Risk:** Low
• Fixed an issue in which connector mode was not set for internal connectors using arcconf.
  • 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 process running symbol was not seen for maxCache device when BPI task was running.
  • Root cause: Progress spinning icon was not added for maxCache when progress was running.
  • Fix: Added the progress spinning icon when the progress is running for maxCache logical device.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which the creation of RAID50 using nine physical devices failed when no legs used.
  • Root cause: Spanned segments for RAID50 was set to 2 always which does not allow a combination of 9 physical devices.
  • Fix: Implemented code to set the spanned segments for RAID 50 based on the device count.
  • Exposure: All previous releases.
  • Risk: Low

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

• Fixed an issue in which the creation of the logical device on 4Kn device with minimum size failed.
  • Root cause: In maxView, the minimum size for a logical for smart family controllers was 16 MB whereas the minimum size for 4Kn devices was 128 MB, irrespective of the block size of devices selected.
  • Fix: Made changes to use 128 MB as minimum size for 4K devices and 16 MB as minimum size for 512K Bytes.
  • 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
• Fixed an issue in which the error message was not removed even though proper capacity was
provided during maxCache creation.
  • Root cause: In maxView, validation for the size entered by the user during maxCache creation
    was present only after clicking the next button.
  • Fix: Made proper changes to add the size related validation in the onKeyUp AJAX event to
    remove the error message on the run.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which maxView cannot match system event SmartPQI to maxView
  information—analysis impossible.
  • Root cause: Missing bus, target, and lun ID information for physical devices which were exposed
to OS.
  • Fix: Added Bus Target Lun information for physical devices which are exposed to OS to maintain
    consistency with the system events.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which the maxView/arcconf was allowed to change the individual connector mode
to RAID when one of the physical device was set as boot device.
  • Root cause: Missing check to block connector mode change to RAID if any of the devices was
    set as boot volume.
  • Fix: Connector mode change to RAID is blocked if any of the physical devices under it is set as
    primary or secondary boot volume.
  • Exposure: All previous releases.
  • Risk: Low

• Fixed an issue in which the maxCache volume could not be created and resulted with an invalid
  message.
  • Root cause: In maxView, while calculating the maximum and minimum size of maxCache, the
    logical double string was not formatted properly according to locale.
  • Fix: The maximum and minimum size of maxCache, is formatted properly according to locale
    to avoid failures.
  • Exposure: All previous releases.
  • Risk: Low

• Added controller-based encryption support for maxView and arcconf.

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.
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.

2.3.4  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 SmartRAID 3100 and SmartHBA 2100 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
maxCache does not support 4Kn drives (SAS or SATA). Also, it can not be assigned to logical volumes consisting of 4Kn drives.

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 SmartHBA 2100/SmartRAID 3100 controller 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 43.
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® SmartRAID 3100 Series and SmartHBA 2100 Series Host Bus Adapters Installation and User's Guide (ESC-2171547)" 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.