Release Notes

SmartHBA 2100 and SmartRAID 3100 Software/Firmware

Released

March 2020

Revision History

Revision	Revision Date	Details of Change
22	March 2020	SR 2.5 Production Release with Firmware 2.66
21	February 2020	SR 2.5.2 Production Release
20	October 2019	SR 2.5 Production Release
19	September 2019	Updated for SR 2.4.8.1 (fw v2.31 Build 0)
18	August 2019	Updated for SR 2.4.8
17	January 2019	SR2.4 Production Release
16	June 2018	SR2.3 Production Release
15	June 2018	Updated for RC Release
14	October 2017	Update supported OSs
13	October 13, 2017	First Production Release
1-12	June 2016-July 2017	Pre-Production Releases.

Contents

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 the following table.

Table 1 • Release Summary

Solutions Release	2.5
Package Release Date	March 18, 2020
Firmware Version	2.66 B0[1,2] (basecode 06.05.002.001)
UEFI Version	1.3.9.2
Legacy BIOS	1.3.9.2
Driver Versions	Windows SmartPQI: • Windows 2012/2016/2019: 106.166.0.1022 • Windows 7/2008: 6.100.0.1014 Linux SmartPQI: • RHEL 6/RHEL 7/RHEL 8/SLES 12/SLES 15: 1.2.10-025 • Ubuntu 16/18: 1.2.10-025 • CentOS 6/7/8: 1.2.10-025 • Debian 8/9: 1.2.10-025 VMware SmartPQI: • VMWare ESXi 6.0/6.5/6.7: 1.0.4.3008 FreeBSD/Solaris SmartPQI: • FreeBSD 11/12: 1.0.4.3008 • Solaris 11: 1.0.4.3008
arcconf/Maxview	B23668

Note:

1. Downgrading to 1.04 B0 or older builds from this release or prior 1.29 releases may cause the board to not boot or have supercap errors due to an incompatibility in SEEPROMs between this release and prior releases. Refer to the section " *Updating the Controller Firmware* " to downgrade an existing board.
2. If the firmware running on the board is older than 0.01 B594, existing data in the logical volumes must be backed up if it needs to be used after the upgrade. After the upgrade from firmware prior to 0.01 B594, the logical volumes will need to be recreated.
3. Only run the driver on firmware 0.01 build 500 or later.

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 SmartHBA 2100/SmartRAID 3100 controller and SmartRAID 3100 and SmartRAID 3100 controller solutions from the Microsemi Web site at *https://storage.microsemi.com/en-us/support/start/*

1.3 Files Included in this Release

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

Firmware Files

Table 2 • Firmware Files

Component	Description	Pre-Assembly Use	Post-Assembly Use
SmartFWx100.bin	Programmable NOR Flash File Use to program NOR Flash for boards that are already running firmware.		X

Table 3 • Firmware Programming Tools

Tool	Description	Executable
Arcconf romupdate	The command allows to upgrade/downgrade the firmware and BIOS image to the controller.	Refer to *Table 7 • Host Management Utilities* on page 4
maxView firmware upgrade wizard	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.	Refer to *Table 7 • Host Management Utilities* on page 4

Driver Files

Table 4 • Windows Storport Miniport SmartPQI Drivers

Package	Drivers	Binary	Version
2012	Server 2019 Server 2016 and Windows 10 Server 2012, R2 and Windows 8.1, 8	SmartPqi.sys	x64
		SmartPqi.inf	x64
		Smartpqi.cat	x64
2008	Server 2008 R2 SP1 and Windows 7	SmartPqi.sys	x64
		SmartPqi.inf	x64
		SmartPqi.cat	x64

Table 5 • Linux SmartPQI Drivers

Drivers	Version
Red Hat Enterprise Linux 8.0/CentOS 8.0	x64
Red Hat Enterprise Linux/CentOS 7.7, 7.6, 7.5[1], 7.4[1], 7.3	x64
Red Hat Enterprise Linux/CentOS 6.10, 6.9[1]	x64
SuSE Linux Enterprise Server 12[1], SP4, SP3, SP2	x64
SuSE Linux Enterprise Server 15, SP1[1]	x64

Drivers	Version
Oracle Linux 7.7 UEK5u2, 7.6 UEK5u2, 7.5 UEK4	x64
Oracle Linux 7.5 with UEK4 (4.1.12-124)	x64
Oracle Linux 7.6 with UEK5u2 (4.14.35)	x64
Oracle Linux 7.7 with UEK5u2 (4.14.35)	x64
Oracle Linux 8.0	x64
Ubuntu 18.04.3 (LTS), 18.04.2, 18.04.1	x64
Ubuntu 16.04.5, 16.04.4	x64
Debian 10	x64
Debian 9.9	x64
Debian 8.11	x64
Citrix xenServer 8.0, 7.6	x64

Note: 1. To mitigate against the Spectre Variant 2 vulnerability, the RHEL 6u9/RHEL7u4/RHEL7u5 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

Drivers	Version
FreeBSD 12.0, 11.3	x64
Solaris 11.3, 11.4	x64
VMware 6.0, 6.5, 6.7	x64

Host Management Software

Table 7 • Host Management Utilities

Description	OS	Executable
ARCCONF Command Line Utility	Windows x64 Linux x64 VMware EXSi 5.5/6.0 XenServer FreeBSD x64 Solaris x86	See the Arcconf download package for the OS-applicable installation executable.
ARCCONF for UEFI		Included as part of the firmware downloadable image.
maxView Storage Manager	Windows x64 Linux x64 VMware EXSi 5.5/6.0	See the maxView Storage Manager download package for the OS-applicable installation executable.

Description	OS	Executable
	XenServer	
maxView vSphere Plugin	vCenter 5.5 and 6.0	See the VMware maxView Storage Manager download package for the OS-applicable installation executable.
Boot USB (offline or pre-boot) for ARCCO-NF and maxView Storage Manager	Linux x64	See the maxView BootUSB download package for the .iso file.

2 What is New?

2.1 Features

The following table lists features supported for this release.

Table 8 • Feature Summary

Feature		Supported in this Release	Future Release
UEFI Driver, Boot Support		X	
Legacy Boot Support		X	
Dynamic Power Management		X	
SMR Drive Support	Enumeration, Unrestricted Command Flow-T-hrough	X	
	SATL Translation for HA/HM SMR Management	X	
	Identify All Drive Types	X	
Driver Support	Windows	X	
	Linux	X	
	VMware	X	
	FreeBSD	X	
	Solaris	X	
	OS certification	X	
Flash Support		X	
MCTP BMC Management		X	
Configurable Big Block Cache Bypass		X	
Green Backup Support for SmartRAID		X	
4Kn Support in RAID		X	

2.2 Fixes

2.2.1 Firmware Fixes

2.2.1.1 Fixes and Enhancements for Firmware Release 2.66 B0

This release includes the following fixes and enhancements:

- Fixed an issue to prevent a potential data inconsistency for RAID 1/10/ADM volumes.

- Root Cause: During background consistency check, an Unrecoverable Read Error (URE) on a drive configured with RAID 1/10/ADM fault tolerant mode may cause a data inconsistency.
- Fix: Modified firmware so a URE will not result in a potential data inconsistency.
- Risk: Low

- Fixed an issue to prevent potential parity inconsistency of a RAID 5/6/50/60 volume.
  - Root Cause: Starting with firmware version 1.98 (SR 2.4), after background parity initialization or operation of a RAID 5/6/50/60 Fault Tolerant volume a potential parity inconsistency may result.
  - Fix: Fixed in previous firmware version 2.30 (SR 2.4.8), background parity initialization or operations will not result in parity inconsistency.
  - Risk: Low

- Fixed a controller 0xFFFFF001 lockup problem while removing all available paths to an enclosure.
  - Root Cause: While disabling both the available paths of an enclosure by hot removing the cables, the internal book keeping structure, which tracks the active I/O module of the enclosure is not updated and index calculation for the expander information data structure becomes invalid and causes a NULL pointer access exception, while background activity is updating LED status of SES target.
  - Fix: Avoid accessing of invalid expander information structure by validating the change in active I/O module when path changes from dual path to single path or single path to none.
  - Risk: Medium

- Fixed a controller 0x1E30 lockup problem during metadata update to a drive which is undergoing device management.
  - Root Cause: When drives are going through device management such as drive firmware update, drive parameters are cleared out in preparation for a reset but if any other thread tries to update the meta data on the same drive, firmware will lockup validating that the drive parameters are initialized.
  - Fix: While a drive is under device management, inform other threads not to update the metadata until device management code completes.
  - Risk: Low

- Fixed a controller hang problem while upgrading/downgrading expander firmware continuously.
  - Root Cause: The allocated memory structures were not freed during the clean-up process of expander firmware update which results in the controller running out of memory and any thread trying to allocate memory will stall resulting in a controller hang.
  - Fix: Free-up the allocated resources during the cleanup process of expander firmware update.
  - Risk: Low

- Fixed a controller 0x1E30 lockup during server reboot/shutdown.
  - Root Cause: When the SmartPQI host transport interface is reset during a server reboot, if there are still outstanding I/Os, firmware will lockup because it has determined there are still outstanding host IOs when the reset was triggered. Firmware incorrectly included commands from out of band transport as well in this outstanding I/Os count that results in the lockup even though there are no outstanding host I/Os.
  - Fix: Do not include outstanding out-of-band (OOB) I/O count while determining outstanding host I/Os during PQI reset and shutdown.
  - Risk: Low

- Fixed MCTP issue where firmware returns more data than expected resulting in buffer overflow at the initiator.
  - Root Cause: When error response IU is sent, data payload was filled with the response IU itself which causes extra data to be sent to initiator from firmware.
  - Fix: For error responses other than under-run cases, firmware should be sending only the error response IU as such. No data should be appended.
  - Risk: Medium

- Fixed a controller hang problem when MCTP initiator sends a Notify on Event command through the out-of-band (OOB) interface.

- Root Cause: OOB requests are not counted as part of logical requests counter in OOB layer, but the code handling Notify on Event directly decrements the submitted counter without considering the host transport type which results in situations where the completed I/O counter is higher than the submitted I/O counter. This causes the controller to be in an indefinite loop causing it to not respond any more.
- Fix: Decrement submitted counter for requests other than OOB requests.
- Risk: Low

- Fixed a controller lockup when server was shut down with controller in an abnormal volume state with an ongoing transformation.
  - Root Cause: When the controller was getting shutdown, the controller hung because shutting down a transformation requires the expand task to indicate that it stopped a complete iteration. Due to the abnormal volume state, the expand task never runs and can never indicate that it is shutdown.
  - Fix: Prevent transformation from happening during the time the transformation is restarting.
  - Risk: Low
- Fixed a problem where a sanitizing drive in a RAID volume was exposed to the host.
  - Root Cause: If the user replaces one of the drives in RAID volume with a drive which was sanitizing offline (when the power is off) and then powers on the server, the volume state is OK and the sanitizing drive was exposed to the host which should not be. The unit state needs to be degraded if the inserted drive is sanitizing and should wait till the sanitize is over. After sanitization, when user enables the drive, rebuild should be started on the drive.
  - Fix: If the sanitizing drive is a part of unit, do not expose the drive to the host.
  - Risk: Low
- Fixed an encryption related problem where master key reset does not complete.
  - Root Cause: When a master key reset is sent by the host application, the new master key name is recorded but the master key reset does not immediately take place. The reset takes place only after the controller has determined that it is safe to do so. One of the safety checks is to ensure that the encryption boot process is complete. In a case, the server was not sending communication requests to the controller. When a controller supports both types of key management modes, the controller was only determining if the encryption boot process was completed from the encryption job requests. Since this never happened the encryption boot process was never set to complete and the master key reset never completed.
  - Fix: Add a check for encryption boot completion whenever the server has booted into an OS.
  - Risk: Low
- Fixed a controller hang during device discovery process when a bad drive is present.
  - Root Cause: When a bad drive is hot plugged, the initial test unit ready commands is not completed, firmware sends IT Nexus Reset that is also not completed by the bad drive. But firmware proceeds to send subsequent test unit ready commands assuming that IT Nexus Reset is complete and these subsequent TUR commands were never sent since the drive is going through reset and firmware waits for these TUR commands completion forever.
  - Fix: If there is no response for IT Nexus Reset for 30 seconds, clear outstanding commands to the drive including resets, which enables the test unit ready commands to be submitted to the drive. Eventually, these TUR commands results in a timeout and the drive is failed.
  - Risk: Medium
- Fixed a controller lockup 0xFFFF001 issue when a drive with old RAID metadata is replaced on a MaxCache volume slot in cold boot window.
  - Root Cause: During controller boot, MaxCache detection logic doesn't properly handle the abnormal volume state when having a drive with old RAID metadata in one of MaxCache SSD slots, which causes a controller lock up.
  - Fix: The controller firmware will properly indicate an unsupported configuration rather than encountering a lock up.
  - Risk: Low

- Fixed a volume marked offline issue due to repeated LUN reset as IOs waited more than 30 seconds to write data into controller cache.
  - Root Cause: For cache-enabled volumes, when the write cache is overloaded, the following events created repeated LUN resets for a volume and the volume was marked offline by the host OS:
    1. I/Os wait for available write cache space for a specific time period, after which the write I/O will be sent to disk. If the I/O sent to the disk encounters an error, it will be put back into the controller cache retry queue. The logic to monitor cache write wait time caused the I/O to have its start timestamp restamped resulting in the I/O to wait longer than 30 seconds.
    2. A Write I/O waiting in the cache retry queue does not have a start time associated to it until it is removed from the retry queue. This results in any accumulated wait time not being taken into account and the I/O can exceed 30 seconds.
    3. Available write cache I/O space distribution to individual volumes had an issue where smaller cache requests gets prioritized over other I/Os, causing a LUN reset.
  - Fix: Additional checks were added to avoid Write cache I/Os start wait time getting restamped; write cache I/Os start wait time is not stamped at the moment when the I/O is queued to the retry queue; cache write space volume distribution logic is refined to accommodate the time associated with an I/O to prioritize them over I/O sizes.
  - Risk: Medium
- Fixed a controller lockup 0xFFFF001 issue when processing I/Os larger than the RAID volume stripe size.
  - Root Cause: When processing I/Os larger than the stripe size, generating the scatter/gather list can result in a NULL memory buffer pointer being allocated, which causes a controller lockup.
  - Fix: The scatter/gather list functionality will check to make sure a memory buffer pointer is correctly allocated to continue processing I/Os.
  - Risk: Low
- Fixed an issue where set cache configuration command fails when reconfiguring MaxCache volume.
  - Root Cause: During MaxCache reconfiguration operation, after all volumes are deleted, one of the counters that tracks the MaxCache presence is not decremented correctly in a minor race condition window.
  - Fix: Two threads responsible for decrementing the MaxCache presence counter when its deleted are now synchronized correctly, allowing the reconfiguration of MaxCache to complete.
  - Risk: Low
- Fixed an issue where activity LED of physical drive keeps blinking even after the volume associated with those drive is moved to a new set of drives.
  - Root Cause: During volume move operation, SES control bits for corresponding drives for activity LED are set correctly, but doesn't get reset when operation completes.
  - Fix: During volume move operation completion, SES control bits for the previous drives are reset correctly.
  - Risk: Low
- Fixed an issue where expand operation disable code is not populated correctly.
  - Root Cause: When expand operation gets suspended due to backup power source related issues, the expand disable code previously set gets overwritten.
  - Fix: Expand disable code value is retained.
  - Risk: Low
- Fixed an issue where set configuration command fails when attempting to create a volume immediately after a drive is hot added into system.
  - Root Cause: When a drive is hot added, the drive present indication to host tools is given early and when set configuration command is sent immediately from host, controller is still fetching the drive size and fails the set configuration command.
  - Fix: During drive hot plug, drive presence indicator is now sent at appropriate time window after controller fetches the required details from the drive for configuration.

- Risk: Low

- Fixed an issue where a drive undergoing sanitize operation cannot be used as a spare drive.
  - Root Cause: When spare drive activation logic finds a replacement drive and is undergoing sanitize operation, it returns incorrect status and prevents the drive to become a spare drive.
  - Fix: Spare drive activation logic is refined to return correct status on sanitizing drive status.
  - Risk: Low

- Fixed a controller hang issue when deleting MaxCache volume.
  - Root Cause: When memory buffer required for a dirty MaxCache data flush operation is not available during MaxCache delete operation, flags tracking the data flush are not cleared, causing the delete operation to fail and the controller to hang.
  - Fix: Flags that track the MaxCache data flush are set after the requested memory is available.
  - Risk: Low

- Fixed an issue updating global drive cache state policy for hot added SATA drives.
  - Root Cause: When a SATA drive gets hot added the validation check points to detect the drive presence returns early and the global drive cache policy doesn't get updated.
  - Fix: Logic for hot added SATA drive presence check is modifed to detect them on time and global drive cache policy changes are applied immediately.
  - Risk: Low

- Fixed an issue where unwanted back up and restore operation occurs, when clearing controller configuration after an unsafe shutdown.
  - Root Cause: During boot after an unsafe shutdown, when controller configuration is cleared, data preservation flags are not cleared, causing unwanted backup and restore operation.
  - Fix: Data preservation flag is now cleared along with controller configuration clear operation.
  - Risk: Low

- Fixed an issue in dual path configuration, when one of redundant path is hot removed, path information is missing in host tools.
  - RootCause: When one path is removed in a dual path configuration, fields related to that path are cleared.
  - Fix: Path information fields are retained even after one of the paths is removed.
  - Risk: Low

- Fixed an issue handling multiple outstanding SMP reset requests from the host and/or higher layer firmware.
  - Root Cause: Lower layer firmware handles multiple resets by queuing the additional resets after the first one. This creates a scenario where if the first reset is lost, subsequent requests end up queued and made to wait till the original (lost) request completes.
  - Fix:

    1. Modify lower layer firmware to reduce the SMP timeout from 50 secs to 10 secs to recover before the upper layers timeout handler kicks in.
    2. Modify lower layer firmware to return a failed response for reset, if it has been outstanding for more than 10 secs so that the upper layer can retry.

    Since a failed response is returned, there will be no resets outstanding in lower layers and the subsequent retries will be processed promptly.

- Fixed an issue with SATA Secure Erase.
  - Root Cause: When a controller SAS PHY connected to a SATA drive is toggled, firmware assumed that the SATA drive was power cycled. But this is not true when the PHY being forced down then up is between the controller and expander to which the SATA drive is connected. So, when the Secure Erase operation is in progress during this PHY toggle test, firmware incorrectly cleared the Secure Erase flag assuming the SATA drive was reset and the command did not complete. In this case, we can assume that the drive will complete the Secure Erase operation and return a completion.

- Fix: Modify firmware to refrain from clearing the Secure Erase flag if the drive did not lose power during a controller PHY toggle.

- Fixed an issue to improve the device reset response time for successful resets.
  - Root Cause: Due to variable expander behavior in handling SMP PHY Control (link/hard reset), after getting the SMP PHY Control response, lower layer firmware waits 10 seconds before deciding if a device has recovered from the reset or not. I/O will be frozen during this time. SATA device LUN reset is translated to device reset, so LUN reset response will not be returned until the 10 second timer has expired. For devices that do not come back after a reset, waiting for 10 seconds before declaring device removal is acceptable. However, for devices that successfully recovered from a reset, it is desirable to declare the reset successful sooner, which results in I/O continuation sooner, and SATA LUN Reset getting a response sooner.
  - Fix: For expander attached device reset (which includes SATA LUN Reset), utilize multiple metrics to detect the device has completed the reset and recovered, and declare the device reset complete sooner.
- Fixed an issue where firmware does not properly detect an over voltage condition when booting with or hot-plugging an over-charged super-cap.
  - Root Cause: Firmware was not gracefully handling situations where controller booted up with an overcharged super-cap or if an overcharged super-cap was hot-plugged.
  - Fix: An over-voltage super-cap is not necessarily unusable so firmware changes were made to immediately proceed to the self-test cycle when detecting an over-charged cap on boot/hot-plug. The self-test cycle includes discharging the cap as part of the capacitance estimating process. If the self-test passes, the cycle will complete with the super-cap charged to an appropriate voltage and the over-voltage condition will be cleared.
- Fixed a lockup issue due to firmware's attempted use of an invalid device handle.
  - Root Cause: An issue was found when a SAS port goes down in very close (ms range) proximity to a drive reset. In this case controller firmware erroneously dereferenced a device handle which was just previously invalidated, causing a lockup.
  - Fix: Modify firmware to add checks to verify if the device handle is valid before using it.

2.2.2 UEFI Fixes

Note: Microsoft signed and secure boot is supported**.**

2.2.2.1 Fixes and Enhancements for UEFI Build 1.3.9.2/Legacy BIOS Build 1.3.9.2

This release includes the following UEFI fixes and enhancements:

- Fixed an issue where platform boot hang was observed while booting with a multi-LUN device connected.
  - Root Cause: Invalid memory access due to mismatch in actual memory allocated and requested size for a SCSI command.
  - Fix: Corrected SCSI request size to match allocated size.
  - Exposure: All previous releases.
  - Risk: Medium
- Fixes an issue where OS boot failed when optimized boot enabled on the platform BIOS.
  - Root Cause: Driver binding supported and start methods has wrong validation for controller handle and remaining device path.
  - Fix: Driver handle and remaining device path validation corrected for driver binding supported and start methods.
  - Exposure: All previous releases.
  - Risk: Low
- Fixed an issue where arcconf EFI shell command line utility prints invalid data.
  - Root Cause: Standard print format specifiers not supported in UEFI EDK2 helper methods.

- Fix: Format specifiers changed to UEFI EDK2 library compatible format.
- Exposure: All previous releases.
- Risk: Low

2.2.3 Driver Fixes

2.2.3.1 Fixes and Enhancements for Linux Driver Build 1.2.10-025

The fixes and enhancements in this release.

- Added support for RHEL7.7 GA, Ubuntu 18.04.3 Final, SLES12SP5 RC1, RHEL8u1 Snapshot 4, and Oracle Linux 7.7 GA.
- Fixed an issue with unique ID for physical devices. The SmartPQI driver exposes a sysfs node named “unique_id” for each device that the driver exposes to the OS. This node contains a 16-byte ID that is supposed to uniquely identify each device. The ID that the driver is returning for physical devices is not necessarily always unique.
  - Root Cause: The driver gets the unique ID from VPD page 83h by reading the 16 bytes at offset 8 of the VPD data. This works for logical devices but does not always work for physical devices.
  - Fix: The driver now gets the unique IDs from the `Report Logical LUNs (RLL) and` Report Physical LUN command (RPL) instead of VPD page 83h.
  - Risk: Low
- Fixed an issue where the TMF timeout is too long (60 seconds).
  - Root Cause: The TMF timeout of 30 seconds is better than 60 seconds.
  - Fix: Shortened the TMF timeout.
  - Risk: Low
- Fixed an issue where controller hang is detected due to loss of sync during a LUN reset TMF to a drive.
  - Root Cause: Since the timeout specified by the host for LUN reset TMF request is infinite, the background thread in controller firmware becomes stuck.
  - Fix: Added support for a timeout on LUN resets. When the controller firmware passes down a LUN reset, it will use the timeout value. If the timeout is hit, the controller firmware will initiate a device reset and upon completion will fail the LUN reset request to the host. The driver will then retry the LUN reset up to 3 times regardless of the failure status.
  - Risk: Medium
- Fixed an issue where controller hangs with INQUIRY command while rebooting and also becomes unresponsive.
  - Root Cause: Currently, there is no timeout mechanism for Inquiry or Pass-through commands.
  - Fix: Introduced new timeout field in RAID IU.
  - Risk: Medium
- Fixed an issue where firmware controller lockup occurs during force reboot. During the force reboot (reboot -f), there are outstanding commands while processing PQI reset, that causes firmware controller lockup.
  - Root Cause: The controller lockup issue was seen when LUN reset and shutdown happens concurrently.
  - Fix: During system shutdown, and before issuing flush cache command, block host IOs from OS and wait for all pending IOs to complete. If there are no pending IO commands, block the lun reset from OS and wait for all the sync commands to complete.
  - Risk: Medium
- Fixed an issue where the `scsi_sysfs_add_sdev` stack trace occurs while adding device.
  - Root Cause: Call trace occurred during device discovery, after multipath target was removed.
  - Fix: `sas_phy_free` should not be called for PHYs that have been set up successfully; rather, use `sas_phy_delete()`.

- Risk: Medium

2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 1.0.4.3008

Following are the fixes and enhancements in this release.

- Fixed an issue of displaying incorrect driver version in UART and arcconf after reboot in FreeBSD 12.0.
  - Root Cause: After the outbox driver installation, the inbox driver was not getting blacklisted.
  - Fix: Added the fix in `pkg-install` and `pkg-deinstall` script to load the outbox driver after reboot.
- Added FreeBSD 12 and 11.3 support.

2.2.3.3 Fixes and Enhancements for Solaris Driver Build 1.0.4.3008

Following are the fixes and enhancements for this release.

- Fixed an issue where the status of the logical drive is not changing to rebuilding when the drive is hot plugged.
  - Root Cause: From the driver `quiesce()`, the PQISRC_SHUTDOWN event was sent in `pqisrc_flush_cache()`. This made the controller firmware to completely shutdown and the background thread to pause in fast reboot(bypassing BIOS/UEFI) environment. This stopped triggering of REBUILDING since background the thread is paused. Also, created issues with OS reboot.
  - Fix: Now PQISRC_NONE_CACHE_FLUSH_ONLY event is passed in `pqisrc_flush_cache()`.

2.2.3.4 Fixes and Enhancements for Windows Build 106.166.0.1022

Following are the fixes and enhancements in this release.

- CSMI RAID support has been added in this release. This feature is supported in all OS versions, except Win7/2008.
- Fixed an issue of memory leak in the Normal Tag table.
  - Root Cause: After successfully allocating normal tag table, SmartPQI driver doesn't clear the 'fUsingSpecialTable' flag. This leads to not freeing the Normal Tag Table memory on driver unload.
  - Fix: Set 'fUsingSpecialTable' flag to FALSE upon successful Normal Tag table allocation
- Fixed the Static Driver Verifier (SDV) defects.
  - Root Cause: Driver is not checking pRecord is a NULL pointer or not after getting a record from MapTraverseTrieStartStop function and it catches in SDV.
  - Fix: Forceful check NULL pointer before accessing device records.
- Added a feature of Timeout Support field in pass-through and task management requests.
- Fixed an issue where HLK sleep test are failing with BSOD.
  - Root Cause: `StorportQueueWorkItem` not able to schedule the worker callback immediately after coming back from sleep state. This issue is only visible when the verifier is enabled on stoport.sys with "Force IRQL Checking" enabled.
  - Fix: For OS targets dynamically changing StorportWorkItem with direct function call during Wake from Sleep/Hibernate. This is a temporary fix and will revert back once Microsoft fixes the issue.
- Fixed an issue where SDDC certification Windows 2016 "PCS-E2ELaunch execute" test case failed with BSOD irql_not_less_or_equal.
  - Root Cause: VoidSrb is taking null arena memory when releasing cmdinfo buffer to arena. After that, immediately the internal controller commands completion is accessing the cmdinfo srb and that leads to BSOD irql_not_less_or_equal.
  - Fix: Added "else-if" condition to prevent accessing Piggy-backed SRB from command info while processing internal controller commands.

2.2.3.5 Fixes and Enhancements for Windows 7/2008 Build 6.100.0.1014

Following are the fixes and enhancements in this release.

- Fixed an issue where the Windows PNP WHQL tests are failing.
  - Root Cause: Unwanted PQI reset triggering the controller post in SIS mode and due to that all the PNP WHQL test cases failed.
  - Fix: PQI reset is moved out from inappropriate place.
- Fixed an issue where the system freezes during repetition of DC Off/On test.
  - Root Cause: There is a polling routine that may fail if the controller takes too long to respond, causing a potential race condition in MemAlloc.
  - Fix: Rearranged the order of these host commands and made them all use polling instead of three of them using interrupts.
- Fixed an issue where the dump file (Memory.dmp) gets corrupted upon reboot.
  - Root Cause: During the SmartPQI Dump mode, Power SRB is completed before the flush cache completion. This causes the system to reboot before the cache flush completes and eventually data in the dump file does not get written to the drive.
  - Fix: Shutdown SRB is completed after the successful Flush Cache operation.

2.2.3.6 Fixes and Enhancements for VMware Driver Build 1.0.4.3008

This release provides the following fixes and enhancements:

- Added Timestamp for debugging TMF issues. For debugging TMF related issues, it is better to have the I/O Timestamp printed when OS issues the TMF. This will help to identify how long the I/O really stayed within the driver.
- Fixed an issue of HBA disk hot reinsert failure.
  - Root Cause: SATA HBA disks represent designator type 3h (locally assigned) which VMware does not support to create unique device id.
  - Fix: When HBA mode disks are hot removed, add them to the removed device list instead of removing these devices from the OS SCSI layer and set the disk to an unavailable state, block any subsequent I/O requests to that device until the disk is reinserted. Also, set the disk re-insertion timeout to make sure to remove the devices that are physically removed with no intention of ever reinserting. Upon reinsertion of the disk within the specified timeout period, mark the device state as available and unblock the I/O to the device, then ask SCSI layer to rescan it to bring that device back to online/active state. If disk re-insertion timeout expires that is, disk not re-inserted within the timeout, then try remove the device permanently by informing OS SCSI layer.
- Added a feature of timeout support field in pass-through and task management requests.
  - Details: SCSI pass through commands that are targeted at devices downstream of the controller can get stuck and never complete. Currently, there is no mechanism to recover a pass-through or TMF LUN Reset command that never completes. Firmware added option to specify the timeout in TMF and RAID request IU.
- Added a feature of timeout for driver initiated inquiry.
  - Root Cause: During the device discovery, driver sends inquiry commands to the discovered devices. These requests don't have any timeout specified. This can result in an inquiry never completing, which causes a firmware task to become suspended indefinitely.
  - Fix: Add timeout for internally framed inquiry commands.

2.2.4 Management Software Fixes

2.2.4.1 Fixes and Enhancements for Arcconf and MaxView Build B23668

This release includes the following fixes and enhancements.

- Fixed an issue where maxView is unable to assign maxCache to a logical drive larger than 280 TB.
  - Root Cause: In maxView, the size limitation of 256 TB for cache line size 64 KB and 1024 TB for cache line size 256 KB was not considered.
  - Fix: Added the size related limitation check for cache line size and provided the default value accordingly. A tool-tip for cache line size provides information about the size constraints.
  - Risk: Low
- Fixed an issue to change the display 'Automatic Failover' to 'Spare Activation Mode' in arcconf getconfig command.
  - Root Cause: arcconf displays a misnomer for 'Spare Activation Mode' as 'Automatic Failover' for Smart Controllers.
  - Fix: Replaced display string 'Automatic Failover' with 'Spare Activation mode'.
  - Risk: Low

2.3 Limitations

2.3.1 Firmware Limitations

2.3.1.1 Limitations for Firmware Release 2.66 B0

This release includes the following firmware limitation:

- Rebuilding task starts very late on the spare drive because over provisioning optimization (OPO) process restarts for each drive failure from RAID group until fault tolerance is reached. This will be seen only on SSD RAID volumes with drives supporting TRIM/UNMAP.
  - **Workaround**: None
- SATA drives attached to a non-Microsemi expander may get into a failed state when upgrading the controller firmware from previous releases to this release due to the expander not clearing STP affiliation.
  - **Workaround**: Power cycle the expanders to clear the STP affiliation.
- When I/Os are performed on drives that respond slowly or which do not respond to READ or WRITE commands, and when Secure Erase is performed on other SATA drives, I/Os become stalled for a period of time. The time the I/Os are paused depends directly on the amount of unflushed data in the cache and speed with which the device responds to error recovery.
  - **Workaround**: None
- A read performance drop is observed on RAID 0 and RAID 1 logical volumes for block sizes 64 KB or higher for the sequential read workloads.
  - **Workaround**: None
- Controller cache will not be converted into 100% read cache, if any backup power source cable error, charge or charge timeout error occurs when expansion or transformation task is active.
  - **Workaround**: None
- Performance drop is observed on certain queue depth for the 4 KB sequential write workload on RAID logical volumes with IOBypass and DDR caching disabled.
  - **Workaround**: Enable the DDR caching for RAID 0 and RAID 1 volumes, to avoid this problem. There are no known workarounds for parity RAID volumes such as RAID 5 or 6.
- When running I/Os on multiple logical drives of various RAID levels, there is a corner-case timing issue where DMA and COMPLETION threads are dead locked and controller triggers 0x1ABD lockup since it does not see requests being completed even after recovery mechanisms.
  - **Workaround**: None

- Encrypted multi-disk RAID 0 and RAID 10 volumes without the DDR cache enabled, can sometimes result in using an incorrect encryption key to encrypt the data under certain workload conditions.
  - Workaround: Enabling the DDR cache, if the controller supports DDR cache, or disabling the encryption feature for the volume

2.3.1.2 Limitations for Firmware Release 1.32 Build 0

- Firmware release 1.32b0 may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations.
  - Description: Refer to entry "Fixed an issue where firmware may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations" in the Firmware fixes section.
  - A fix for this issue is available in the 1.60 B0 firmware release. If a firmware flash failure is occurring, try the following workarounds:
    - *Workaround:* If there are no target devices (expanders or drives) attached to the controller, attach a target device to the controller and try the host management operation again.
    - *Workaround*: If the system is operating using UEFI, the HII tool can be used to flash the firmware to this release as outlined in the *Microsemi SmartIOC 2100/SmartROC 3100 Installation and User's Guide (ESC-2170577),* appendix entry “Updating the SmartIOC 2100/SmartROC 3100 Controller Firmware”.
    - *Workaround*: If there are target devices attached to the controller and this issue occurs or none of the workarounds can be used, contact Microsemi Support.

2.3.2 UEFI Limitations

2.3.2.1 Limitations for UEFI Build 1.3.9.2 /Legacy BIOS Build 1.3.9.2

There are no known limitations for this release.

2.3.3 Driver Limitations

2.3.3.1 Limitations for Linux Driver Build 1.2.10-025

This release includes the following limitation:

- Occasionally, the Linux driver might fail to unload. This limitation will occur if maxview services are running in the background and the user tries to unload (disable) the Linux SmartPQI driver with the `rmmod` command.
  - Workaround: In such a scenario, execute the following commands:
    - `service stor_redfishserver stop`
    - `service stor_tomcat stop`
    - Unload driver

2.3.3.2 Limitations for Windows Driver Build 106.166.0.1022

This release includes no known limitations for Windows Driver Build 106.166.0.1022, for Windows 2012/2016/2019.

2.3.3.3 Limitations for Windows 7/2008 Driver Build 6.100.0.1014

There are no known limitations for this release.

2.3.3.4 Limitations for FreeBSD Driver Build 1.0.4.3008

There are no known limitations for this release.

2.3.3.5 Limitations for Solaris Driver Build 1.0.4.3008

This release includes the following Solaris driver limitations:

- UEFI Secure boot is not supported.
- Deleting a Logical device through arcconf when I/O is running on Logical device, deleted logical devices are still showing on top of OS when # format or # echo | format command is executed to check the devices. Few drives are removed successfully and few others are not. This is observed for some of the device removal when OS API ndi_devi_offline() returns with failure.

2.3.3.6 Limitations for VMware Driver 1.0.4.3008

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 B23668

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 controller boards to the latest release.

If the controller is currently running 1.60 b0 firmware or newer, follow these steps:

1. **Mandatory:** Flash the target with the provided " SmartFWx100.bin" image with arcconf/maxView software.
2. **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 16.
2. **Mandatory:** If flashing completes, cold boot the system to refresh all components.

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.

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
2355 W. Chandler Blvd.
Chandler, AZ 85224 USA

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

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

ESC-2161026