Release Notes

SmartHBA 2100 and SmartRAID 3100 Software/Firmware

Released

March 2020

Revision History

Revision	Revision Date	Details of Change
23	March 2020	SR 2.5.2 Production Release with Firmware 2.93
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	2.5.2
Package Release Date	March 21, 2020
Firmware Version	2.93 B0[1,2] (basecode 06.05.003.000)
UEFI Version	1.3.10.2
Legacy BIOS	1.3.10.2
Driver Versions	Windows SmartPQI: • Windows 2012/2016/2019: 106.178.0.1009 • Windows 7/2008: 6.102.0.1026 Linux SmartPQI: • RHEL 6/RHEL 7/RHEL 8/SLES 12/SLES 15: 1.2.12-025 • Ubuntu 16/18: 1.2.12-025 • CentOS 6/7/8: 1.2.12-025 • Debian 8/9: 1.2.12-025 VMware SmartPQI: • VMWare ESXi 6.0/6.5/6.7: 1.0.4.3017 FreeBSD/Solaris SmartPQI: • FreeBSD 11/12: 1.0.4.3017 • Solaris 11: 1.0.4.3017
arcconf/Maxview	B23699

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/CentOS 8.1, 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], SP5, SP4, SP3, SP2	x64
SuSE Linux Enterprise Server 15, SP1[1]	x64

Drivers	Version
Oracle Linux 7.5 with UEK4u7 (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 with UEK5	x64
Ubuntu 19.01	x64
Ubuntu 18.04.3, 18.04.2, 18.04.1	x64
Ubuntu 16.04.5, 16.04.4	x64
Debian 10	x64
Debian 9.9	x64
Citrix xenServer 8.0, 7.6	x64
Fedora 30	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 11.4, 11.3	x64
VMware 6.7 U3/U2/U1, 6.5 U3/U2/U1, 6.0 U3	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	
Out of Band interface selection support of MCTP or PBSI		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 2.93 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 hang issue when maxCache flush task encounters multiple UREs.
  - Root Cause: When maxCache flush task chooses a dirty cache page which has URE, the read will fail as expected due to URE and the cache page will be reinserted back to the head to flush queue. This causes the maxCache flush task to be in a loop that ends up in controller hang.
  - Fix: MaxCache flush task will now reinsert the dirty cache page with URE into the tail of the flush queue and fix the controller hang condition.
  - Risk: Low
- Added an enhancement so that the next background consistency check iteration will be restarted when the consistency check fail bit is set and will not wait for 14 days.
- Fixed a data integrity/miscompare problem due to memory buffer overrun issue when extending the size of a 4kn volume.
  - Root Cause: During completion of volume size extension, while wiping out last 1 MiB of extended volume, firmware zeros out the memory buffer beyond what it has allocated because it did not take into consideration the number of 4kn blocks in the 1 MiB calculation. Depending on nature of corrupted buffers, impact would be seen on the functionality for which those buffers are intended for. In this case, impacted buffers were always the ones used for XOR calculation to regenerate the failed drive data in a degraded parity volume which resulted in data corruption.
  - Fix: Calculated the correct number of blocks to zero out taking the drives native block count into consideration.
  - Risk: low
- Fixed an issue where after a controller dirty shutdown with maxCache volume, controller clear configuration does not clear the maxCache unit information in controller RAID metadata.
  - Root Cause: After controller dirty shutdown, controller clear configuration commands incorrectly bypasses MaxCache units that ran into dirty shutdown condition.
  - Fix: Clear configuration command handling is modified to clear the MaxCache unit information independent of clean/dirty shutdown state.
  - Risk: Low
- Fixed an issue where clear configuration commands fails when maxCache volume has multiple UREs within a maxCache page block size.
  - Root Cause: When MaxCache has multiple UREs within a configured MaxCache page block, error handling logic sets a busy flag that clear controller config operation checks and fails the command.

- Fix: URE error handling is modified to not set the busy flag more often so controller clear configuration command completes.
- Risk: Low

- Fixed an issue where a controller would be stuck in survival mode when no temperature sensors were included in the design.
  - Root Cause: During boot, firmware expects to find at least one temperature sensor as required by the hardware reference design. If, for some reason, no temperature sensors are included in the design the firmware would incorrectly use other sensor configuration data to configure the survival mode temperature thresholds. This results in the board falling immediately into survival mode.
  - Fix: When attempting to set up the survival mode thresholds at boot, validate the sensor data being used is associated with a temperature sensor.
  - Risk: Very Low
- Fixed an issue where an installed failed drive could lead to "Clear Configuration" failures.
  - Root Cause: Firmware is correctly determining a drive should be marked failed when it is unable to access the media to check for RAID metadata presence, but it was unable to manage write failures to clear that metadata. This issue could also occur with a drive currently undergoing SCSI Sanitize since the metadata cannot be read from or written to during that process.
  - Fix: Do not attempt to clear RAID metadata from drives for which firmware could read the metadata at discovery.
  - Risk: Low
- Fixed issues with a spurious drop in performance under sequential workloads.
  - Root Cause: The incoming IO path was considering some non-IO requests as a part of the workload. Because these are not IO requests, when the coalescing layer inspected the target LBA's it would determine the sequential stream had stopped and would then subsequently stop coalescing the host IO's into full stripe operations. This transition to non-full stripe operations and then later back to coalescing full stripes would appear to the host as a momentary decrease in performance. These momentary dips in performance would occur as frequently as the injected non-IO requests under the active host workload.
  - Fix: Modified the incoming host request request parsing logic so that non-IO requests are no longer considered by the coalescing logic.
  - Risk: Low
- Fixed a controller failure (hang or lockup) when running concurrent traffic to HBA and RAID targets in mixed controller mode.
  - Root Cause: An HBA target request not queued via the hardware accelerated queuing path was being incorrectly considered in the RAID volume I/O coalescing layer. This results in a controller failure because the I/O was incorrectly placed on a list of "last parsed I/O" reserved for RAID volume I/O.
  - Fix: Do not place non -RAID volume I/O onto the "last parsed I/O" list since these are not applicable to the RAID volume logic.
  - Risk: Low
- Fixed an issue where a bad drive could cause a controller to not be found or an OS to not boot after a reboot.
  - Root Cause: During discovery of the drive after a reboot, the controller firmware was repeatedly attempting to start the drive but the drive was repeatedly reporting 04/xx/xx check conditions (HARDWARE ERROR) to TestUnitReady commands. Parallel to this, the controller has a limit on the amount of time afforded to topology discovery and if that time has been exceeded then the UEFI/BIOS driver or OS driver will give up waiting for the controller to become ready. In this case, the error/recovery logic for these devices with fatal errors was causing this timeout to be exceeded.
  - Fix: Fail a drive immediately rather than attempt repeated error recovery after it reports the first HARDWARE ERROR status on TestUnitReady because these devices are not expected to suddenly become good again.

- Risk: Low

- Fixed an issue where a host request is incorrectly returned as failed for a DEGRADED volume when a replacement drive has just been inserted.
  - Root Cause: There was a race condition regarding the replacement drive's status in which the hot plug logic would have accepted the newly added device's state as "OK" and in-progress error handling logic for recovery/regenerative I/O path logic looking at this same status before the drive has started to rebuild. In this case, an I/O in recovery was being abandoned when it should have proceeded without involvement of the newly added drive.
  - Fix: The logic to determine whether an I/O is recoverable in the error recovery path was corrected to look at the state of the RAID volume's data column rather than the discovery state of the drive.
  - Risk: Low
- Fixed an issue where modifying the spare configuration under an active spare rebuild would cause rebuild to halt.
  - Root Cause: The volume configuration change was causing the rebuild to prematurely stop and volume state to be promoted to OK.
  - Fix: This type of configuration change request will now be rejected until the spare is no longer in active use; for example, until after rebuild has completed.
  - Risk: Low
- Fixed an issue where direct-attach drives not attached to a backplane where no enclosure management schema is defined results in incorrect device location reporting.
  - Root Cause: When there is no enclosure management schema defined for a connector, e.g. not SGPIO or some I2C schema, and a direct-attach drive is attached, then code review identified a scenario where this type of device connectivity would result in incorrect device location reporting due to association with a non-existent virtual device associated with the enclosure management schema.
  - Fix: Removed logic that was blindly associating devices in this situation with the non-existing enclosure management target.
  - Risk: Low
- Fixed an issue where device configuration settings applied for RAID volumes were not being applied if the device transitioned from HBA usage to RAID usage during run-time under mixed mode.
  - Root Cause: The device discovery for boot had portions of logic not being applied to devices at configuration time. This would result in some device settings such as the PER (Post Error) setting in the "Read/Write Error Recovery" mode page from being saved to the desired states if an HBA target was re-configured to be a RAID target.
  - Fix: Created a process to re-apply these settings at configuration time.
  - Risk: Low
- Fixed an issue where RAID volume re-appear after "Clear Configuration" operation and reboot are executed.
  - Root Cause: The firmware was not attempting to clear stale metadata from drives previously marked as failed. If the previously failed drive(s) become functional enough to read that metadata on the next boot, firmware would trust that it is valid and use it.
  - Fix: Attempt to clear metadata during "Clear Configuration" from all drives even if they have been previously marked.
  - Risk: Low
- Fixed a controller lockup issue when toggling one of the path to an enclosure in dual domain configuration.
  - Root Cause: During enabling/disabling one of the path to an enclosure in dual domain configuration, an incorrect timeout value is set before the path can be added back.
  - Fix: Conditions that set the incorrect timeout value is modified to have the correct value, allowing the path to be added back.
  - Risk: Low

- Fixed a problem where rebuild does not start on a logical drive created out of SSD drives within 1200 seconds.
  - Root Cause: After failing one drive within a sub-group of a RAID 10, 50, or 60 SSD logical volume, RPI is started on the first available spare. Before RPI ends on this spare, if another drive from a different sub-group of the same logical volume fails, firmware is deactivating the spare through internal book keeping which is resulting in re-starting of RPI on the same spare drive.
  - Fix: Do not deactivate the currently assigned spare if RPI/OPO is already running on that spare drive.
  - Risk: Low
- Fixed a performance drop problem on sequential read with cache enabled volumes and the following I/O size cases:
  - 16k-64k when there are <= 4 logical volumes in the configuration
  - 16k-256k, with queue depths of 1 and 2
  - Root Cause: Deeper read ahead logic was disabled in firmware release 1.98 build 0 to address a performance problem that subsequently resulted in a drop in the above-mentioned cases. It is found out that having read ahead during each cache hit helps configurations with a small number of logical volumes and workloads with low queue depths.
  - Fix: Bring back the read ahead logic during each cache hit. Read ahead algorithm is optimized further to avoid CPU cycles by determining if necessary amount data is already read ahead and bail out early from cache code.
  - Risk: Low
- Fixed a controller lock up issue during background cache flush.
  - Root Cause: During DDR cache flush, a race condition between two threads (host write and cache flush for same LBA range) in a small timing window causes probable out of order disk write operation which controller detects and lockups.
  - Fix: Added thread synchronization mechanism for the small timing window to streamline the disk writes from two threads.
  - Risk: Low
- Fixed an issue where wrong connector information is displayed for SATA drives after a cable hot plug event on dual I/O module enclosure configuration.
  - Root Cause: In dual I/O module enclosure configuration with SATA drives, when one of the cables is hot removed/added the connector information is not updated properly leading to wrong connector information to be exported for display in host tools.
  - Fix: Active connector information is updated correctly for SATA drives when cable is hot added on dual I/O module enclosure configuration.
  - Risk: Low
- Fixed an issue where failed drives are physically present in its slot are not reported correctly in installed drive map.
  - Root Cause: When a drive fails and is physically present in its slot, corresponding installed drive map bits are not updated to reflect the correct status to be exported for display in host tools.
  - Fix: Added more conditions to check for failed drive and its PHY active status to correctly set the installed drive map bit.
  - Risk: Low
- Fixed an issue where device drops are observed after multiple create/delete drive zone group configuration commands.
  - Root Cause: When multiple create/delete drive zone group configurations are performed, an incorrect out of bound device index value is assigned and causes devices to drop out of configuration.
  - Fix: Device index is correctly reinitialized during multiple create/delete drive zone group configurations.
  - Risk: Low

- Fixed an issue where status LED doesn't blink during sanitize operation on expander attached drives configuration.
  - Root Cause: When drives behind expander undergo sanitize operation, controller updates the appropriate bit in SES pages to blink the status LED, but the command fails to reach expander due to incorrect sanity checks.
  - Fix: Sanity checks are refined to allow the SES page update for the drives undergoing sanitize operation behind expander to blink the status LEDs.
  - Risk: Low
- Fixed an issue where cache disable code is not updated properly when backup power source charging is timed out
  - Root cause: When the green backup unit goes into failed state due to power source charging timeout, firmware is not updating the cache disable code accordingly. This also results in firmware logs capturing incorrect information on power source status.
  - Fix: Cache disable code is set to appropriate value.
  - Risk: Low
- Fixed an issue where all zeros response was sent for ATA PASSTHROUGH commands, such as SMART READ DATA, through Out of Band (OOB) host transport.
  - Root cause: For OOB transport commands transfer length is calculated based on incoming CDBs solely and firmware was parsing the CDBs based on 16 byte/12 byte SCSI CDB which resulted in incorrect transfer length. In addition, when completing OOB commands back to host, firmware was not copying data back for ATA Passthrough since the command direction was set incorrectly.
  - Fix: Firmware has added support in OOB host transport path for parsing the ATA PASSTHROUGH commands' transfer length based on T10 SAT specification. Command direction is calculated based on T_DIR bit incoming CDBs for ATA PASSTHROUGH commands for copying the data back to host accordingly.
  - Risk: Low

2.2.2 UEFI Fixes

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

2.2.2.1 Fixes and Enhancements for UEFI Build 1.3.10.2/Legacy BIOS Build 1.3.10.2

This release includes the following UEFI fixes and enhancements:

- Added a new menu item "Modify Cache Settings". Cache related options are moved from the Modify Controller Settings menu to the Modify Cache Settings menu. Options for No Battery Write Cache and Cache Ratio are moved to the Modify Cache Settings menu. Warning message is displayed in the form for applicable cache options.
- Added option to configure OOB (PBSI/MCTP) settings under Modify Controller Settings.
- Fixed an issue where MaxCache Array creation failed for SSD RAID0 Logical drive.
  - Root Cause: When the first editable logical drive has an accelerator of none, the driver will force the read cache ratio to 100% read and 0% write. This causes the MaxCache association to fail.
  - Fix: Walk through all logical drives to determine if the cache is enabled before determining the final values of the read and write cache ratio.
  - Risk: Low
- Fixed an issue where the Disk Utilities menu shows controller as additional device.
  - Root Cause: Improper logic for filtering controller device from other physical devices.
  - Fix: Corrected logic for filtering controller device from other physical devices.
  - Risk: Low

This release includes the following Legacy BIOS fixes and enhancements:

- Fixed an issue where CTRL A disk utilities shows wrong device port, box, and bay.
  - Root Cause: Check was missing for `alt_paths_phys_conn[0]`.
  - Fix: Verify `alt_paths_phys_conn[0]` against 00 and if it is equal over write the content with --.
  - Risk: Low
- Fixed an issue with incorrect array name when array count exceeds 26 in legacy BIOS.
  - Root Cause: Only one of the two characters were being used to display the array name
  - Fix: Use both the characters while printing the array name.
  - Risk: Low

2.2.3 Driver Fixes

2.2.3.1 Fixes and Enhancements for Linux Driver Build 1.2.12.025

This release includes the following enhancements:

- Added support for the following: RHEL8u2 Beta 1, RHEL7u8 Snapshot-3, SLES15SP2 Snapshot-2, SLES 12 SP5 GMC, and BCLinux 7u6 ARM only.
- Fixed an issue where when Secure Memory Encryption (SME) is enabled, the smartPQI driver doesn't work, which results in the failure of kernel boot because it fails to allocate PQI error buffer.
  - Root Cause: The coherent DMA mask value caused the driver to fall back to Software Input Output Translation Lookaside Buffer (SWIOTLB) when SME is active.
  - Fix: For correct operation, call the `dma_set_mask_and_coherent()` to properly set the mask for both streaming and coherent, to inform the kernel about the device's DMA addressing capabilities.
  - Risk: Medium
- Fixed an issue where there is a performance degradation when the default maximum transfer size for rotating media is used. The smartPQI firmware bypasses the controller cache for any request greater than 1 MB impacting the I/O performance. However, there may be cases where larger transfer sizes are required.
  - Root Cause: The maximum transfer size can be greater than 1 MB due to kernel patch d2be537c3ba3 applied to kernels v4.3 and later. This depends on logical volume configuration and stripe size.
  - Fix: Added a module parameter (LV xfer limit) to limit logical volume transfer size to 1 MB.
  - Risk: Medium
- Add ID support for new entry level SmartRAID controllers

2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 1.0.4.3017

Following are the fixes and enhancements for this release.

- Add ID support for new entry level SmartRAID controllers.
- Fixed an issue where OS crashes during physical drive hot removal. The physical drive hot removed while heavy I/O is running on it from FIO and that leads to OS crash.
  - Root Cause: Driver is not waiting for physical drive I/O completions before wiping out the device memory.
  - Fix: Waiting for outstanding commands using atomic operations with maximum of 30 seconds interval.
    - Set IOBypass enable flag to false, if the driver receives IOBypass response error with status as IOBYpass path disabled. The flag will stop the further in flight I/O's in IOBypass path.
  - Risk: Medium

- Fixed an issue where the driver is currently processing "`softs->max_outstanding_io -1`" outstanding commands instead of "`softs->max_outstanding_io`" commands.
  - Root Cause: The driver is currently processing one command less than the maximum outstanding IO commands.
  - Fix: Added condition to process all outstanding commands which includes:
    1. Releasing used tags after completing outstanding commands. While unloading a driver it is not necessary but it should be cleaner to have it.
    2. Assigning NULL pointer to device after freeing up memory.
  - Risk: Low
- Fixed an issue where FreeBSD crashes while issuing firmware test lockup during I/Os.
  - Root Cause: Driver was not clearing the RCB structure after I/O completion. When controller went offline, driver was using the RCB structure values to complete the pending I/Os. Since the RCB fields were not cleared, there was an I/O double completion and leading to OS crash.
  - Fix: Reset the RCB before completing I/Os.
  - Risk: Low
- Fixed an issue during dynamic unload of SmartPQI driver.
  - Root Cause: Controller is in complete shutdown while unloading (or) detaching a driver. While unloading a driver, the cache flush is called with PQISRC_SHUTDOWN event type in smartpqi_shutdown function and that leads to complete shutdown of the controller. Hence, the next driver load fails.
  - Fix: During driver unload, we are calling Cache flush event type with PQISRC_NONE_CACHE_FLUSH_ONLY. This allows dynamically load of driver.
  - Risk: Medium

2.2.3.3 Fixes and Enhancements for Solaris Driver Build 1.0.4.3017

Following are the fixes and enhancements for this release.

- Add ID support for new entry level SmartRAID controllers.
- Fixed an issue where OS might crash during physical drive hot removal. The physical drive got removed while heavy I/O is running on it from FIO and that leads to OS crash.
  - Root Cause: Driver is not waiting for physical drive I/O completions before wiping out the device memory.
  - Fix: Waiting for outstanding commands using atomic operations with maximum of 30 seconds interval.
    - Set IOBypass enable flag to false, if the driver receives IOBypass response error with status as IOBypass path disabled. The flag will stop the further in flight I/O's in IOBypass path.
  - Risk: Medium

2.2.3.4 Fixes and Enhancements for Windows Build 106.178.0.1009

- Added ID support for new entry level SmartRAID controllers.
- Fixed an issue where the SRB conversion routine failed to properly identify bidirectional flags which meant it would always fall back to read.
  - Root Cause: An if was used when instead of "else if".
  - Fix: Replaced second if with "else if" in order to ensure the first if, which checks whether bidirectional is used when both IN and OUT flags are set.
  - Risk: Medium. Assumes the RAID stack knows how to handle bidirectional flags.
- Fixed an issue where a bogus CDB in IOCTL SAS Passthru caused unssafe memcpy and double completion at host side.

- Root Cause: Forgot to add return statement on this error condition.
- Fix: Added the return statement. Since user provided an unsupported IOCTL, the request cannot be continuously parsed.
- Risk: Low

- Fixed an issue where the DP WLK—hot replace test is failing; the multi-tag table memory was not getting released.
  - Root Cause: Since OFA was introduced and the multi-tag table is now tied to the operational queues it must always recreate operational queues.
  - Fix: Rework code to always rebuild operational queues and a new multi-tag table. Also, always preserve Records, per lun memory, RAID map data, and I/O error buffer.
  - Risk: Medium
- Fixed an issue where the Static Driver Verifier (SDV) reports null check defects.
  - Root Cause: Driver is not checking if 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.
  - Risk: Low
- Fixed an issue related to the static variables. Since the static variables are global to all controller instances, these counters should have been made local to the device extension rather than system-wide.
  - Root Cause: Used static instance rather than controller instance.
  - Fix: Used controller context for tracking this information.
  - Risk: Low
- Fixed an issue where system would crash when doing unnecessary initialization of the multi-tag table after declaring controller lockup.
  - Root Cause: Unnecessary initialization of the multi-tag table.
  - Fix: Removed unnecessary initialization when declaring a controller lockup.
  - Risk: Low

2.2.3.5 Fixes and Enhancements for VMware Driver Build 1.0.4.3017

Following are the fixes and enhancements for this release:

- Added ID support for new entry level SmartRAID controllers.
- Fixed an issue with controller offline status not being detected after triggering a controller lockup.
  - Root Cause: Driver is not detecting controller offline status after triggering a controller lockup when RBOD is connected. Current SmartPQI driver uses common Timer queue for all the timers—heartbeat, rescan and host wellness update. When one of the timer callbacks is stuck, the others will not get a chance to run. Here the rescan timer was executing and the controller went offline (because of test lockup). Rescan timer function was waiting for the response from the firmware. Since the controller is locked up, driver will not get any response. Controller lockup is not getting detected by the driver since the heartbeat timer is not getting chance to run.
  - Fix: Added new timer queue for heartbeat timer.
  - Risk: Low
- Fixed an issue where firmware was getting lockup during SOB replay.
  - Root Cause: Heartbeat timer handle will run in each 5 seconds and issue firmware lockup if there is no heartbeat update for 5 seconds. During "sob_replay", firmware updated the heartbeat after 8 seconds. This results in NMI issued from the driver.
  - Fix: Increase the heartbeat timeout to 10 seconds.
  - Risk: Low
- Fixed an issue where device memory is not getting freed in some cases and preventing driver unload after array creation/deletion.

- Root Cause: Some commands fail due to hot removal of devices. OS detected this issue ahead of the driver and it invoked the driver routine to free the path. Driver will not free the device since driver rescan did not happen. Later, driver detected the device removal and requested OS to free the path. Since the path was already destroyed by the OS, it will not invoke the driver routine to free the path data structure and this resulted in a memory leak.
- Fix: If OS invoked the routine to destroy the path, free the device memory in the driver rescan itself.
- Risk: Medium

2.2.4 Management Software Fixes

2.2.4.1 Fixes and Enhancements for Arcconf and maxView Build B23699

This release includes the following fixes and enhancements.

- Fixed an issue where setting connector mode does not work when user provides all connectors as input in arcconf.
  - Root Cause: When the user tries to set the connector mode of all connectors to the same mode, the user specified connector mode is compared against the controller mode, instead of the connector mode which causes the operation to fail.
  - Fix: Added changes to compare with existing connector's mode when user tries to set the connector mode of all connectors.
  - Risk: Low
- Fixed an issue where an active spare is not displayed as part of group segment of a logical drive.
  - Root Cause: Active spare information is not added for a logical device segment information.
  - Fix: Added changes to display Active spare information segment information if the 'missing' device has associated spare to it.
  - Risk: Low
- Added support for SmartRAID Entry Level controller.
- Added support to configure PBSI/MCTP: arcconf provides support to switch the BMC interface between PBSI and MCTP along with the other related configuration changes. arcconf can also reset or disable the BMC interface. A power cycle will be needed to reflect the newly set BMC interface settings.

2.3 Limitations

2.3.1 Firmware Limitations

2.3.1.1 Limitations for Firmware Release 2.93 B0

This release includes the following firmware limitations:

- Configurations using the ASR-3162/i/e controller may, in rare cases, lockup during self test of an unused hardware block.
  - Workaround: None
- Configurations using SMR drives may encounter a firmware lockup when the Report Zones command fails to complete due to SAS link error(s).
  - Workaround: Fix the link errors to avoid this problem.
- When creating a logical volume with SSDs using Rapid Parity Initialization (RPI), if one of the SSDs stops responding to Test Unit Ready commands, after 30 seconds of the RPI process the controller may hang.
  - Workaround: Use Background Parity Initialization instead of RPI to avoid the issue.
- If redundant data can't be regenerated during a host write request on a degraded logical volume due to bad blocks on all data drives, the respective LBA will still be marked bad, but it will not be returned

with error status. A subsequent read to these LBAs will result in a medium error with sense data (KCQ 03/11/00) because the block is already marked bad by firmware.
    - 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
- 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.
- Logical array undergoing an expansion or array type transformation may not complete properly (including controller lockup) if its migrated from one controller to another when transformation/expansion is active. Once transformation/expansion activity is complete, array can be migrated to another controller.
    - 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 1.3.10.2/Legacy BIOS Build 1.3.10.2

There are no known limitations for this release.

2.3.3 Driver Limitations

2.3.3.1 Limitations for Linux Driver Build 1.2.12.025

There are no known limitations for this release.

2.3.3.2 Limitations for Windows Driver Builds 106.178.0.1009

There are no known limitations for this release.

2.3.3.3 Limitations for FreeBSD Driver Build 1.0.4.3017

There are no known limitations for this release.

2.3.3.4 Limitations for Solaris Driver Build 1.0.4.3017

There are no known limitations for this release.

2.3.3.5 Limitations for VMware Driver Build 1.0.4.3017

This release includes the following VMware driver limitation:

- OS upgrade from VMware 6.5 and 6.7 with Smartpqi driver 1.0.4.3017 to VMware 7.0 is not supported in 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 B23699

This release includes the following limitations:

- With Arcconf CLI, creation of a RAID60 on an existing array may fail intermittently with error message “Not enough space in array 0 to create logicaldrive”, when sufficient space is available in the array.
  - Workaround: Execute the create RAID60 command again.
- In Windows Server 2019, on continuous RAID configuration management operation, maxView may lose its communication with the maxView Redfish service due to an exception in the maxView Redfish server module while accessing the event log file.
  - Workaround: Restart the ‘maxView Redfish Server' service using Windows services application.

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