HBA 1200 Software/Firmware Release Notes

© 2021 Microchip Technology Inc.
and its subsidiaries

Table of Contents

1. About This Release....3
   1.1. Release Identification....3
   1.2. Files Included in this Release....3
2. What's New?....6
   2.1. Features....6
   2.2. Fixes and Enhancements....7
   2.3. Limitations....14
3. Updating the Controller Firmware....16
   3.1. Updating Controllers to Latest Firmware....16
4. Revision History....17
The Microchip Website....18
Product Change Notification Service....18
Customer Support....18
Microchip Devices Code Protection Feature....18
Legal Notice....19
Trademarks....19
Quality Management System....20
Worldwide Sales and Service....21

1. About This Release

The development release described in this document includes firmware, OS drivers, tools, and host management software for the HBA 1200 solutions from Microchip.

1.1 Release Identification

The firmware, software, and driver versions for this release are shown in the following table.

Table 1-1. Release Summary

Solutions release	3.1.4
Package release date	August 10, 2021
Firmware version	3.01.04.072
UEFI/Legacy BIOS	1.4.3.6/1.4.3.2
Driver versions	**Windows Drivers**: • Windows 2019, 2016, Windows 10: 1010.6.0.1025 **Linux SmartPQI**: • RHEL 7/8: 2.1.12-055 • SLES 12/15: 2.1.12-055 • Ubuntu 18/20/21: 2.1.12-055 • Oracle Linux 7/8: 2.1.12-055 • Citrix Xenserver 8: 2.1.12-055 • Debian 9/10: 2.1.12-055 • CentOS 7/8: 2.1.12-055 **VMware**: • VMware ESX 6/7: 4150.0.119 **FreeBSD/Solaris**: • FreeBSD 11/12/13: 4130.0.1008 • Solaris: 11: 4120.0.1005
ARCCONF/maxView	B24308

1.2 Files Included in this Release

This section details the files included in this release.

Table 1-2. Firmware Files

Component	Description	Pre-Assembly Use	Post-Assembly Use
SmartFWx200.bin	Production-signed programmable NOR Flash File. Use to program NOR Flash for boards that are already running firmware.		X

Table 1-3. Firmware Programming Tools

Tool	Description	Executable
ARCCONF	ARCCONF CLI Utility	ARCCONF BXXXXX.zip
maxView	maxView Utility	MAXVIEW XXX BXXXXX.zip

Driver Files

Table 1-4. Windows Drivers

OS	Version
Server 2019, 2016, Windows 10	x64

Table 1-5. Linux Drivers

OS	Version
RHEL 8.4, 8.3, 8.2, 8.1, 7.9, 7.8, 7.7	x64
CentOS 8.3, 8.2 ,8.1 ,8.0 ,7.9 ,7.8 ,7.7	x64
SLES 12 SP5, SP4	x64
SLES 15 SP3, SP2, SP1	x64
Ubuntu 20.04.2, 20.04.1, 20.04, 18.04.5, 18.04.4	x64
Ubuntu 21.04	x64
Oracle Linux 8.3, 8.2, 7.9, 7.8, UEK6U1 (5.4.17-2036)	x64
Oracle Linux 8.2 UEK R6	x64
Debian 10.5, 9.13	x64
Fedora 33 (inbox)	x64
XenServer 8.2	x64

Table 1-6. FreeBSD, Solaris, and VMware Drivers

OS	Version
ESX6.5U3/U2	x64
ESX6.7U3/U2	x64
ESX7.0U2/U1	x64
FreeBSD 13, 12.2, 11.4	x64
Solaris 11.4	x64

Host Management Software

Table 1-7. maxView and ARCCONF Utilities

Description	OS	Executable
ARCCONF Command Line Utility	Windows x64 Linux x64 VMware 6.5 and above XenServer UEFI support	See the arcconf_B#####.zip for the installation executables for the relevant OS.
maxView Storage Manager	Windows x64 Linux x64 VMware 6.5 and above XenServer UEFI support	See the maxview_linux_B#####.zip, maxview_win_ B#####.zip,and the maxview_vmware_B#####.zip for the installation executables.
maxView vSphere Plugin	VMware 6.5 and above	See the maxview_vmware_B#####.zip for the installation executables.
Boot USB (offline or pre-boot) for ARCCONF and maxView Storage Manager	Linux x64	See the maxview_offline_bootusb_B#####.zip for the .iso file.

2. What's New?

This section shows what’s new in this release.

2.1 Features

The following table lists the features supported for this release.

Table 2-1. Features Summary

Features		Supported in this Release	Future Release
UEFI driver, boot support		X	
Legacy boot support		X	
Dynamic power management		X	
Driver support	Windows	X	
	Linux	X	
	VMware	X	
	FreeBSD	X	
	Solaris	X	
	OS certification	X	
Flash support	ARCCONF utility	X	
maxView tool support		X	
ARCCONF tool support		X	
MCTP BMC management		X	
4Kn support in RAID and HBA		X	
Controller-based encryption (CBE) support[1]		X	
Out-of-band interface selection support of MCTP or PBSI		X	
VPP Backplane support		X	
PBSI support		X	
Configurable Expander SSU settings		X	

Note:

1. Only available for encryption-enabled products.

2.2 Fixes and Enhancements

This section shows the fixes and enhancements for this release.

2.2.1 Firmware Fixes

This section shows the firmware fixes and enhancements for this release.

2.2.1.1 Fixes and Enhancements for Firmware Release 3.01.04.072

This release provides the following fixes and enhancements.

- Improved the serial log output for attached device inventory. Specifically, improved the information regarding NVMe device unique identifiers and separated/re-organized several groups of information to improve readability.
- Added support to report a unique SCSI ID for NVMe devices via InquiryVPD 83h and new CISS style ReportPhysicalLUNs formats to support OS drivers in collecting this information.
- Added direct-cabling functionality for cables compatible with the 'auto-detect' logic.
- The "PMS (Performance Monitoring Statistics)" metrics API has been deprecated. Several performance-related metrics in the "M&P" API have also been deprecated; however, error-related counters are still being maintained.
- NVMe device PCIe identifiers are now reported via IdentifyPhysicalDevice command. This supports PLDM reporting as a part of adding SCSI unique identifiers for NVMe devices.
- Added support for firmware to report new OS driver device inventory command formats in support of future support for unique SCSI device information for NVMe devices and also added support to report a unique SCSI ID for NVMe devices via Inquiry VPD 83h and new CISS style ReportPhysicalLUNs formats to support OS drivers in collecting this information.
- Fixed an issue preventing hot-added drive discovery with an SGPIO backplane if that backplane segment initially had no drives installed at boot.
  - *Root cause*: The logic which decides to disable 'unused' PHYs incorrectly evaluated ports attached to an SGPIO backplane as not being associated with a backplane. Because SGPIO does not have a standardized out-of-band mechanism to detect hotplug, disabling the controller PHYs after initial discovery prevented future hotplugs from being detected.
  - *Fix*: For a direct-attach backplane, leave the controller PHYs enabled to support in-band hot-plug detection.
  - *Risk*: Low
- Fixed an issue where the reported link rate for NVMe drives is incorrectly reported when the desired port width is not the same as the actual port width linked up.
  - *Root cause*: The reporting from firmware was not granular enough to report the disparity. After that issue had been resolved, the firmware would try to disable 'unused' PHYs from the port group, but this would cause the entire port to the drive to be disabled.
  - *Fix*: Firmware logic was updated to report the discrete PHY link status/rates correctly. The firmware logic to disable 'unused' PHYs was also modified to not disable PHYs for an NVMe port in which at least one PHY is linked up.
  - *Risk*: Low
- Fixed an issue where the controller WWID is printed incorrectly during boot up to the UART/SOB log.
  - *Root cause*: A refactoring change to the general manufacturing format outputs was made alongside addition of support for a separate new format. This refactored code incorrectly treats the WWID as a QWORD field instead of a BYTE array which results in the output appearing to be endian swapped. The change only impacted this printing function—the WWID was otherwise properly used and reported elsewhere appropriately.
  - *Fix*: Changed the logic back to printing the WWID value as a BYTE array.
  - *Risk*: Low
- Fixed an issue where drives attached to a VPP backplane have incorrect bay numbers and are presented to the host incorrectly.
  - *Root cause*: The VPP discovery logic found a TWI device at address 0xAE but read all 0xFF's from it. This resulted in behavior that set up the enclosure assuming a bad EEPROM, but this resulted in multiple customer experience issues.
  - *Fix*: The logic that follows having a bad EEPROM was adjusted to account for a responsive TWI target with no valid data. This case will now be treated more like a direct-cabled case without an enclosure.

  - *Risk*: Low
- Fixed an issue where NOR flash corruption in the firmware version or checksum table results in uncorrectable corruption.
  - *Root cause*: The firmware version information was being corrected but not marked as such, and the checksum table was not being corrected. In both cases, the controller continues to report that it detected and did not repair the corruption.
  - *Fix*: Both of these cases are correctable if the redundant image is coherent, so logic was added to perform the correction as well as properly report that status.
  - *Risk*: Low
- Fixed an issue where the first cold boot after a firmware update reports redundant image corruption.
  - *Root cause*: DDR training results are stored in a redundant image section such that they can be referenced to speed up training during boot. On firmware update, these results are cleared to allow potentially new algorithms in the new firmware to establish new/better results. The RAID stack was incorrectly including this section in comparisons between the active and inactive image contents which was triggering a false image corruption message.
  - *Fix*: Do not include this section in the image comparison unless the stored results are already coherent.
  - *Risk*: Low
- Fixed a potential 0x1ABD controller lockup when a SATA drive is being failed with IO outstanding and the drive fails to respond to Identify Device after reset.
  - *Root cause*: There is a possibility in firmware to queue an IO to a device at a time when it is being failed and this can lead to the IO being unrecoverable as the drive becomes unresponsive. Another device reset may have recovered the IO, however the firmware IO timeout logic explicitly excluded sending recovery task management to devices already marked failed.
  - *Fix*: In timeout handling, allow actions such as device reset against devices marked failed. Also added an active IO recovery step to the device failure routine to actively abort outstanding requests rather than waiting for the device (or SATL) to process it's queues normally.
  - *Risk*: Low
- Fixed an issue where firmware incorrectly reported "Online firmware Activation" as a supported feature set of this product.
  - *Root cause*: Firmware incorrectly advertised Online Firmware Activation functionality as supported. When host software observes these support bits and attempts to use the feature, it may encounter errors because it is not actually supported.
  - *Fix*: Modified the various feature reporting mechanisms to indicate this feature is not supported.
  - *Risk*: Low
- Fixed a TLB exception lockup issue when multiple Out-of-Band MCTP requests were sent to the firmware at the same time.
  - *Root cause*: TLB Exception/NULL pointer exception occurs in the firmware when it receives asynchronous MCTP requests at the same time in a session when a previous MCTP request has not been processed fully. Due to this, the firmware gets into a timing sensitive situation where one of the threads in the firmware is setting up the packetized MCTP responses by accessing the OOB session memory buffer which was just freed up by another thread responsible for processing MCTP requests. This is because the firmware handles one MCTP request in a session in the synchronous manner, if it receives another request from the same session before completing the existing request, it deletes the old session context and starts processing the new request.
  - *Fix*: To gracefully handle this situation, the firmware will use spinlock while accessing the OOB session from different threads.
  - *Risk*: Low
- Resolved a problem with reduced performance when the host is submitting large sequential IO streams at high queue depth.
  - *Root cause*: When requests are being coalesced and staged, this activity occurs in either the host IO ingress context (PARSE) or in the RAID mapping context (MAPPER). When PARSE is making decisions about when to stage data and staging resources are constrained, it was entering a busy-wait loop to allow completions to free resources and would check this loop every 10 ms (or 100 IO/s). In this particular

workload, the steady-state of the system was causing the IO ingress to be completely gated on this loop which resulted in a very predictable and fixed amount of IO to occur.
  - *Fix*: The busy-wait loop timer was reduced to 100 µs (or 10k IO/s) which is more than sufficient to saturate the throughput of the controller.
  - *Risk*: Low

2.2.2 UEFI/Legacy BIOS Fixes

This section shows the UEFI/Legacy BIOS fixes and enhancements for this release.

2.2.2.1 Fixes and Enhancements for UEFI Build 1.4.3.6/Legacy BIOS Build 1.4.3.2

This release provides the following fixes and enhancements.

- Added an HII option in the port discovery protocol settings to support cable attached drives.
- Added support for Drive Last Failure reason status in the HII disk information menu.
- Fixed an issue where failed HBA drives are not shown in HII.
  - *Root cause*: Failed HBA devices are not displayed in HII and driver health messages.
  - *Fix*: Populate and provide available information on failed devices in HII and driver health messages.
  - *Exposure*: All previous versions.
  - *Risk*: Low
- Fixed an issue where the UEFI Self Certification Tests SCT fails for Component name2 protocol.
  - *Root cause*: GetControllerName of Component name2 protocol does not validate input language. SCT fails when incorrect language is provided as input.
  - *Fix*: Supported language validation added for GetControllerName of Component name2 protocol.
  - *Exposure*: All previous versions.
  - *Risk*: Low
- Fixed an issue where the port discovery protocol changes do not provide the status to inform users that a reboot is required.
  - *Root cause*: Port discovery protocol operation status only shows if it is success or failed.
  - *Fix*: Added reboot required message in final status of port discovery protocol settings.
  - *Exposure*: All previous versions.
  - *Risk*: Low
- Fixed an issue where the UEFI ARCCONF CLI produces an error as an unrecognized command in the EFI shell.
  - *Root cause*: Incorrect header for sense feature page commands leading to wrong feature bit validation treating the ARCCONF CLI feature as not supported.
  - *Fix*: Corrected sense feature page command headers as per specification to obtain correct feature bits for ARCCONF CLI feature.
  - *Exposure*: All previous versions.
  - *Risk*: Low

2.2.3 Driver Fixes

This section shows the driver fixes and enhancements for this release.

2.2.3.1 Windows Driver Fixes

This section shows the Windows driver fixes and enhancements for this release.

2.2.3.1.1 Fixes and Enhancements for Windows Driver Build 1010.6.0.1025

This release provides the following fixes and enhancements.

- Fixed an issue where the OS would possibly fail to boot.
  - *Root cause*: Driver can fail to load because Writing Administrator Queue Configuration Function Register and then reading register without delay can give erroneous stale status.
  - *Fix*: Added a 1ms=1000us delay after writing Administrator Queue Configuration Function Register, but before polling begin polling status.
  - *Risk*: Low

2.2.3.2 Linux Driver Fixes

This section shows the Linux driver fixes and enhancements for this release.

2.2.3.2.1 Fixes and Enhancements for Linux Driver Build 2.1.12-055

This release provides the following fixes and enhancements.

- Fixed an issue where duplicate device nodes for Ultrium tape drive and medium changer are being created.
  - *Root cause*: The Ultrium tape drive is a multi-LUN SCSI target. It presents a LUN for the tape drive and a 2nd LUN for the medium changer. Our controller firmware lists both LUNs in the RPL results. As a result, the smartpqi driver exposes both devices to the OS. Then the OS does its normal device discovery via the SCSI REPORT LUNS command, which causes it to re-discover both devices a 2nd time, which results in the duplicate device nodes. This broken behavior was masked by an earlier smartpqi bug that caused the OS to skip its device discover for this type of device. This masking bug was fixed by a recent change to smartpqi to report more accurate information about SAS initiator port protocols and target port protocols.
  - *Fix*: When the OS re-discovers the two LUNs for the tape drive and medium changer, the driver recognizes that they have already been reported and blocks the OS from adding them a second time.
  - *Risk*: Low
- Fixed an issue where in some situations when the driver takes the controller offline, a kernel crash can occur.
  - *Root cause*: While taking controller offline, it is possible for the driver to fail IOs which have already been completed by the OS, causing a kernel crash.
  - *Fix*: If the device has been marked offline by the OS, do not fail IOs pertaining to that device because IOs may have been previously completed.
  - *Risk*: Low
- Fixed an issue with device removal using sysfs.
  - *Root cause*: Defining slave_destroy causes SML to call into our slave_destroy to remove the device from SCSI table. Our slave_destroy is not complete.
  - *Fix*: Remove slave_destroy.
  - *Risk*: Low
- Fixed an issue where request_irq failed during system hibernation.
  - *Root cause*: The first argument irq in "request_irq" is not correct.
  - *Fix*: If the interrupt mode is being set to INTx, use PCI device's "irq" as first parameter to request_irq().
  - *Risk*: Low
- Fixed an issue where during system hibernation, driver frees all the irqs, disables MSIx interrupts and requests legacy INTx interrupt. When driver invokes request_irq(), OS returns—EINVAL. For example, smartpqi 0000:b3:00.0: irq 191 init failed with error -22 genirq: Flags mismatch irq 34. 00000080 (SmartPQI) vs. 00000000 (i40e-0000:1a:00.0:misc).
  - *Root cause*: The first argument irq in request_irq is not correct
  - *Fix*: : If the Interrupt mode is being set to INTx, use PCI device's irq as first parameter to request_irq().
  - *Risk*: Low
- Due to a change in the SCSI mid-layer, some Linux distributions may take a long time to come up if the system is rebooted while a hard disk(s) is being sanitized. This has been observed on RHEL 7.9/RHEL8.3 and SLES 15SP2.
  - *Root cause*: During boot-up, some OSes appear to hang when there are one or more disks undergoing sanitize. According to SCSI SBC4 specification section 4.11.2 Commands allowed during sanitize, some SCSI commands are permitted, but read/write operations are not. When the OS attempts to read the disk partition table a CHECK CONDITION ASC 0x04 ASCQ 0x1b is returned which causes the OS to retry the read until sanitize has completed. This can take hours.
  - *Fix*: Add in a Test Unit Ready to HBA disks and do not present them to the OS if 0x02/0x04/0x1b (sanitize in progress) is returned.
  - *Risk*: Low
- Fixed an issue with request leakage, performance drop, and system crash.
  - *Root cause*: The issue happens in a max configuration where heavy I/O load is exercised with occasional LUN resets on the exposed devices. While failing queued IOs in the TMF path, there was a request leak and hence stale entries in request pool with reference count being non-zero. In the shutdown path, there is a BUG_ON to catch stuck I/O either in the firmware or in the driver. The unfreed stale request caused

system crash. If the above situation keeps occurring then the I/O request pool keeps leaking and there could be a significant performance drop.
  - *Fix*: The driver now frees the leaked request properly in the TMF path while failing outstanding requests.
  - *Risk*: Low
- Fixed an issue to avoid failing IOs for devices which are not online.
  - *Root cause*: While taking controller offline, it is possible for the driver to fail IOs which have already been completed by the OS, causing a kernel crash.
  - *Fix*: If the device has been marked offline by the OS, do not fail IOs pertaining to that device since IOs may have been previously completed.
  - *Risk*: Low

2.2.3.3 VMware Driver Fixes

This section shows the VMware driver fixes and enhancements for this release.

2.2.3.3.1 Fixes and Enhancements for VMware Driver Build 4150.0.119

This release provides the following fixes and enhancements.

- Added support for the new extended formats in the data returned from the Report Physical LUNs command for controllers that support this feature. The new formats allow the reporting of 16-byte WWIDs.
- Fixed an issue where PSOD was observed while running MBT tool.
  - *Root cause*: During attach(), driver saves private structure pointer for each adapter in a global array. Index to the array was never decremented during unload. This resulted in out of bound access of array and leads to PSOD.
  - *Fix*: Clear the private structure pointer during driver detach().
  - *Risk*: Medium
- Fixed an issue with SG element alignment only for tri-mode controllers.
  - *Root cause*: NVMe requires four byte alignment attribute in DMA engine settings.
  - *Fix*: For tri-mode controllers, use a customized dma engine with 4-byte DMA SG alignment parameter set.
  - *Risk*: Medium
- Fixed an issue to avoid failing IOs for devices which are not online.
  - *Root cause*: While taking controller offline, it is possible for the driver to fail IOs which have already been completed by the OS, causing a kernel crash.
  - *Fix*: If the device has been marked offline by the OS, do not fail IOs pertaining to that device since IOs may have been previously completed.
  - *Risk*: Low
- Fixed an issue with an unsafe device quiesce process.
  - *Root cause*: An OS API for synchronizing and flushing interrupts was not being called.
  - *Fix*: Add call to the OS API for flushing any pending interrupts.
  - *Risk*: Low
- Fixed an issue with ESXi PSOD in Smartpqi TMF handler.
  - *Root cause*: During "virtual reset" TMF, driver iterates through IO structures and will issue aborts for all pending IOs. While framing an abort request, the driver uses the device structure pointer from the IO structure. If IO associated with the IO structure completes in parallel, the device structure pointer might reset to NULL, which will result in a page fault.
  - *Fix*: Use device structure pointer given by the OS TMF handler.
  - *Risk*: Low
- Fixed an issue with ESXi PSOD due to page fault.
  - *Root cause*: An inquiry command to one of the drives is timing out and the OS issues a TMF abort. During TMF completion, the driver will print the TMF status. Internally, this uses the driver private structure which was not set when framing the TMF request.
  - *Fix*: Set driver private structure pointer when framing TMF request.
  - *Risk*: Low
- Fixed an issue with ESXi PSOD due to Heartbeat NMI.

  - *Root cause*: Driver acquires a lock to get a slot on the inbound queue. All cores might end up in using the same inbound queue if the number of SCSI completion worlds are less than the number of cores. In most cases, the number of SCSI completion worlds are the same as number of sockets and most servers have 1 to 2 sockets. This might cause lock congestion as many threads will be trying to acquire the same lock. Driver uses a custom lock that does a tight busy wait if the lock is not available. This will cause the IO submission thread to hold the CPU core and the ESXi heartbeat thread might not get a chance to run for a long time. This will result in ESXi issuing an NMI and PSOD the server.
  - *Fix*: Use the spinlock in submission path.
  - *Risk*: High
- Fixed an issue where the system hangs during driver load.
  - *Root cause*: Driver uses an infinite timeout for sending internal commands related to event configuration during driver init stage.
  - *Fix*: Added timeout for sending internal commands related to event configuration during driver init stage.
  - *Risk*: Low
- Fixed an issue with ESXi 7.0 u2 PSOD while booting.
  - *Root cause*: Smartpqi driver creates maximum of 64 outbound queues. Queues are created based on number of cores/scsi completion worlds and MSIX availability. At max, driver will create 64 queues and 64 handlers should be registered. Driver handler data array size was 63 instead of 64 and resulting in PSOD.
  - *Fix*: Corrected the handler data array size.
  - *Risk*: Low
- Fixed an issue where abort messages would flood logs during device reset tests.
  - *Root cause*: All IO requests pending to a device will be aborted by an incoming device reset request. For devices capable of high queue depths, this could be tens or hundreds of individual abort requests, per device reset.
  - *Fix*: Change logging level for this type of message from WARN to INFO, so that it is only printed when someone purposefully changes the driver's logging level to do debug or analysis.
  - *Risk*: Low
- Fixed an issue with excessive logging during device resets.
  - *Root cause*: When resets are executing, incoming IO requests are blocked and returned with status DEVICE BUSY. A message is printed to warn that the device is undergoing a reset.
    When many device resets are occurring, such as during reset certification testing, this generates a large volume of logging activity and can cause logs to be frequently archived.
  - *Fix*: Change this message to be logged only when logging level is specifically changed to a level at or above INFO (0x6). The messages were set to log at or above WARN (0x2), and driver logging level is NOTE (0x3) by default.
  - *Risk*: Low
- Fixed an issue with PSOD during driver unload.
  - *Root cause*: Smartpqi driver maintains a linked list of hot-removed devices. Whenever a new device is present, driver checks whether that device is already present in the remove_device_list, and if it is present, driver moves that device from remove_device_list to actual device list. Entries in the remove_device_list will be reviewed in fixed time interval and list will be updated by removing device which has been in that list for more than 20 minutes (to handle vSAN hotplug test). During driver unload, driver checks for any devices present in the list and does the cleanup (free the device memory). PSOD stack trace indicates an invalid device memory freeing during this cleanup.
  - *Fix*: Remove entry from the device list whenever the device memory is freed up.
  - *Risk*: Medium
- Fixed an issue to remove erroneous status messages.
  - *Root cause*: Status message is reporting iu_type instead of status.
  - *Fix*: Remove status field from messaging.
  - *Risk*: Low
- Corrected queue depth setting for physical device.
  - *Root cause*: Driver gets the queue depth value from firmware for each target. If firmware does not give a valid queue depth value for a target, driver is sets queue depth to a default value (1014 for LD, 27 for

PD). But for all physical devices, current driver resets the queue depth to maximum queue depth (1014) irrespective of whether firmware gave a valid QD or not.
  - *Fix*: Add proper check while setting the device queue depth.
  - *Risk*: Low
- Fixed an issue where the driver produces too much debug logging.
  - *Root cause*: Driver's default logging level was set very high during development phase, and was never readjusted for production use.
  - *Fix*: Change default log level back to normal default level, 3. Adjust logging level of some functions as needed.
  - *Risk*: Low
- Fixed an issue with verbose logging from error handlers.
  - *Root cause*: Debug messages used in driver's error handling functions are being printed at “normal” system logging level.
  - *Fix*: Use recently-added controller flag and compile-time option to turn off the unwanted messaging.
  - *Risk*: Low
- Fixed an issue with PSOD while printing DMA memory tag.
  - *Root cause*: Driver uses tag (a string) to identify the DMA memory and it was maintained by using a char pointer. This tag is assigned while allocating the DMA memory. In some places, tag was defined as local char array and driver was maintaining pointer to that. During driver unload, when driver tries to print the tag, that resulted in page fault as tag memory was local to allocation function.
  - *Fix*: Maintain tag using char array instead of keeping tag address.
  - *Risk*: Low

2.2.3.4 FreeBSD/Solaris Driver Fixes

This section shows the FreeBSD/Solaris driver fixes and enhancements for this release.

2.2.3.4.1 Fixes and Enhancements for FreeBSD Driver Build 4130.0.1008

This release provides the following fixes and enhancements.

- Fixed an issue when drives are added/removed/offline, HBA devices, and controllers are displayed as a default RAID 0 value.
  - *Root cause*: There is no check for physical devices or controllers before printing display info.
  - *Fix*: Modify the messaging so that it prints differently based on physical devices and controllers to identify them accordingly.
  - *Risk*: Low
- Fixed an issue where uninitialized CCB structure causes undefined behavior when it is shared with the CAM layer.
  - *Root cause*: CCB is being used without clearing stack values.
  - *Fix*: Clear CCB before it is used.
  - *Risk*: Low
- Fixed an issue in which the driver is disabling drives if it detects the controller going offline but there is no information that logs the controller lockup code when it is offline.
  - *Root cause*: The driver is not displaying the controller lockup code.
  - *Fix*: Display the lockup code in driver logs. Also, the timer handler is disabled when the controller is offline to prevent a system crash in the event of delay during post memory deletion.
  - *Risk*: Low

2.2.3.4.2 Fixes and Enhancements for Solaris Driver Build 4120.0.1005

This release provides the following fixes and enhancements.

- Fixed an issue when drives are added/removed/offline, HBA devices, and controllers are displayed as a default RAID 0 value.
  - *Root cause*: There is no check for physical devices or controllers before printing display info.
  - *Fix*: Modify the messaging so that it prints differently based on physical devices and controllers to identify them accordingly.

- *Risk*: Low

2.2.4 Management Software Fixes

This section shows the management software fixes and enhancements for this release.

2.2.4.1 maxView Storage Manager/ARCCONF Fixes

This section shows the maxView Storage Manager/ARCCONF fixes and enhancements for this release.

2.2.4.1.1 Fixes and Enhancements for maxView Storage Manager/ARCCONF Version 2.0.0 Build 24308

This release provides the following fixes and enhancements.

- Support to add firmware event log buffer as part of "Savesupportarchive".
- Support for controller to report failed physical devices in the configuration.
- Fixed an issue where remote ARCCONF has OpenSSL security vulnerabilities.
  - *Root cause*: Remote ARCCONF uses older version of open source library OpenSSL which had security vulnerabilities.
  - *Fix*: Added changes to Remote ARCCONF by adding the latest version of OpenSSL library that had addressed the security vulnerabilities.
  - *Risk*: Low
- Fixed an issue where maxView does not display the configuration properly when a physical device has a model name with quotation marks ["] in it.
  - *Root cause*: Having quotation marks [“] in the physical device model name has corrupted the JSON format of the configuration making maxView unable to display it properly.
  - *Fix*: Added changes to JSON configuration creation to address characters such as [“].
  - *Risk*: Low

2.3 Limitations

This section shows the limitations for this release.

2.3.1 Firmware Limitations

This section shows the firmware limitations for this release.

2.3.1.1 Limitations for Firmware Release 3.01.04.072

There are no known limitations for this release.

2.3.2 UEFI/Legacy BIOS Limitations

This section shows the UEFI/Legacy BIOS limitations for this release.

2.3.2.1 Limitations for UEFI Build 1.4.3.6/Legacy BIOS Build 1.4.3.2

There are no known limitations for this release.

2.3.3 Driver Limitations

This section shows the driver limitations for this release.

2.3.3.1 Windows Driver Limitations

This section shows the Windows driver limitations for this release.

2.3.3.1.1 Limitations for Windows Driver Build 1010.6.0.1025

There are no known limitations for this release.

2.3.3.2 Linux Driver Limitations

This section shows the Linux driver limitations for this release.

2.3.3.2.1 Limitations for Linux Driver Build 2.1.12-055

This release includes the following limitations.

- On AMD/RHEL 7.9 systems, the system might panic due to the a bug in the IOMMU module. For details, refer to https://lore.kernel.org/linux-iommu/20191018093830.GA26328@suse.de/t/
  - *Workaround*: Disable the IOMMU setting option in BIOS.
- Depending on hardware configurations, the smartpqi expose_ld_first parameter may not always work consistently.
  - *Workaround*: None
- Hibernating Linux system using ”pm-hibernate” command causes system to hang.
  - *Workaround*: None

2.3.3.3 VMware Driver Limitations

This section shows VMware driver limitations for this release.

2.3.3.3.1 Limitations for VMware Driver Build 4150.0.119

There are no known limitations for this release.

2.3.3.4 FreeBSD/Solaris Driver Limitations

This section shows FreeBSD/Solaris driver limitations for this release.

2.3.3.4.1 Limitations for FreeBSD Driver Build 4130.0.1008

There are no known limitations for this release.

2.3.3.4.2 Limitations for Solaris Driver Build 4120.0.1005

There are no known limitations for this release.

2.3.4 Management Software Limitations

This section shows management software limitations for this release.

2.3.4.1 maxView Storage Manager/ARCCONF Limitations

This section shows the maxView Storage Manager/ARCCONF limitations for this release.

2.3.4.1.1 Limitations for maxView Storage Manager/ARCCONF Version 2.0.0 Build 24308

There are no known limitations for this release.

3. Updating the Controller Firmware

This section describes how to update the controller firmware to the latest release.

3.1 Updating Controllers to Latest Firmware

If running firmware is 3.01.00.006 or lower, please contact Adaptec Apps team at ask.adaptec.com.

3.1.1 Upgrading to 3.01.04.072 Firmware

1. For controllers running 3.01.02.042 or higher firmware, flash with 3.01.04.072 version of firmware "SmartFWx200.bin" provided in this package using maxview or ARCCONF utility.
2. Power cycle the server.

4. Revision History

Table 4-1. Revision History

Revision	Date	Description
B	08/2021	Updated for SR 3.1.4 release.
A	06/2021	Document created.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2021, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-8477-6

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Australia - Sydney**	**India - Bangalore**	**Austria - Wels**
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	**China - Beijing**	**India - New Delhi**	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	**Denmark - Copenhagen**
Fax: 480-792-7277	**China - Chengdu**	**India - Pune**	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	**China - Chongqing**	**Japan - Osaka**	**Finland - Espoo**
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	**China - Dongguan**	**Japan - Tokyo**	**France - Paris**
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	**China - Guangzhou**	**Korea - Daegu**	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	**Germany - Garching**
Fax: 678-957-1455	**China - Hangzhou**	**Korea - Seoul**	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Haan**
Tel: 512-257-3370	**China - Hong Kong SAR**	**Malaysia - Kuala Lumpur**	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	**Germany - Heilbronn**
Westborough, MA	**China - Nanjing**	**Malaysia - Penang**	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	**Germany - Karlsruhe**
Fax: 774-760-0088	**China - Qingdao**	**Philippines - Manila**	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	**Germany - Munich**
Itasca, IL	**China - Shanghai**	**Singapore**	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	**China - Shenyang**	**Taiwan - Hsin Chu**	**Germany - Rosenheim**
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	**China - Shenzhen**	**Taiwan - Kaohsiung**	**Israel - Ra'anana**
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Fax: 972-818-2924	**China - Suzhou**	**Taiwan - Taipei**	**Italy - Milan**
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	**China - Wuhan**	**Thailand - Bangkok**	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	**Italy - Padova**
Houston, TX	**China - Xian**	**Vietnam - Ho Chi Minh**	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	**Netherlands - Drunen**
Indianapolis	**China - Xiamen**		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	**China - Zhuhai**		**Norway - Trondheim**
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			**Poland - Warsaw**
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			**Romania - Bucharest**
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			**Spain - Madrid**
Tel: 951-273-7800			Tel: 34-91-708-08-90
Raleigh, NC			Fax: 34-91-708-08-91
Tel: 919-844-7510			**Sweden - Gothenberg**
New York, NY			Tel: 46-31-704-60-40
Tel: 631-435-6000			**Sweden - Stockholm**
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			**UK - Wokingham**
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			
Fax: 905-695-2078			