SmartRAID 3200 and SmartHBA 2200 Software/Firmware Release Notes

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?....5
   - 2.1. Fixes and Enhancements....5
   - 2.2. Limitations....16
3. Updating the Controller Firmware....21
   - 3.1. Updating Controllers to Latest Firmware....21
4. Revision History....22

Microchip Information....23
- Trademarks....23
- Legal Notice....23
- Microchip Devices Code Protection Feature....24

1. About This Release

The release described in this document includes firmware, OS drivers, tools, and host management software for the SmartRAID 3200 and SmartHBA 2200 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	3.4.2
Package release date	December 4, 2024
Firmware version	3.01.33.44
UEFI/Legacy BIOS	2.16.4/2.16.3
Driver versions	**Windows Drivers**: • Windows 2025, 2022, 2019, Windows 11, 10: 1016.10.0.1004 **Linux SmartPQI**: • Rocky Linux 9: 2.1.32-035 • RHEL 7/8/9: 2.1.32-035 • SLES 12/15: 2.1.32-035 • Ubuntu 20/22/24: 2.1.32-035 • Oracle Linux 7/8/9: 2.1.32-035 • Citrix Xenserver 8: 2.1.32-035 • Debian 11/12: 2.1.32-035 **VMware**: • VMware ESX 7.0/8.0: 4704.0.108 **FreeBSD**: • FreeBSD 14/13: 4570.0.1006
ARCCONF/maxView	4.23.00.27147
PLDM	6.45.7.0

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 2025, 2022, 2019, Windows 11, 10	x64

Table 1-5. Linux Drivers

OS	Version
RHEL 9.5[1] (inbox only), 9.4, 9.3, 8.10, 8.9, 7.9	x64
SLES 12 SP5	x64
SLES 15 SP6, SP5	x64
Ubuntu 20.04.6, 20.04	x64
Ubuntu 24.04.1, 24.04, 22.04.5, 22.04.4, 22.04	x64
Oracle Linux 7.9 UEK6U3	x64
Oracle Linux 9.4, 9.3, 8.10, 8.9, UEK7U2	x64
Debian 12.6, 11.10	x64
Fedora 40 (inbox)	x64
Citrix XenServer 8.2.1	x64
Rocky Linux 9.4, 9.3	x64
SLE-Micro 6.0, 5.5 (inbox only)	x64

Note:

1. New OS is minimally tested with inbox driver. Full support is expected in the next release.

Table 1-6. FreeBSD and VMware Drivers

OS	Version
ESX 8.0 U3/U2, 7.0 U3/U2	x64
FreeBSD 14.1, 13.3	x64

Note:

1. New OS is minimally tested with inbox driver. Full support is expected in the next release.

Host Management Software

Table 1-7. maxView™ and ARCCONF Utilities

Description	OS	Executable
ARCCONF Command Line Utility	Windows x64 Linux x64 VMware 7.0 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 7.0 and above XenServer	See the maxview_linux_B#####.zip, maxview_win_ B#####.zip,and the maxview_vmware_B#####.zip for the installation executables.
maxView™ vSphere Plugin	VMware 7.0 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 Fixes and Enhancements

This section shows the fixes and enhancements for this release.

2.1.1 Firmware Fixes

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

2.1.1.1 Fixes and Enhancements for Firmware Release 3.01.33.44

This release includes the following fixes and enhancements:

- Added support for transferring controller Serial Output Buffer (SOB) log using PLDM Type 7 command.
- Added support to ignore unsupported drive attached to UBM backplane.
- Added support to reduce UEFI load time.
- Added support to allow configuring internal connector's PCIe PHY rate.
- Added support to log device information that caused a 0x1ABx lockup in the controller event log.
- Fixed an issue that a lockup occasionally happens when RAID 0 Predictive Spare Rebuilding (PSR) starts.
  - *Root cause*: When RAID 0 Predictive Spare Rebuilding (PSR) starts, firmware checks all the outstanding host requests if they have an internal lock acquired. If not, firmware will wait for all the requests to be returned to host before starting the PSR. Firmware mistakenly checks an internal request which sometimes might look like an outstanding host request. The internal request would never be released, which would cause firmware to wait there forever. The long wait blocks the firmware's background thread and causes the lockup.
  - *Fix*: Firmware doesn't check the internal requests.
  - *Risk*: Low
- Fixed an issue where continuous prints were observed in the UART in the presence of an Otherwise Owned Locked SED drive.
  - *Root cause*: Since Otherwise Owned SED is in locked state, any IOs accessing the drive will cause the SED to return a Small Computer System Interface (SCSI) error. Firmware displays the drive error when processing the failure.
  - *Fix*: To reduce the number of errors reported, firmware will print the error in a decreasing frequency over time until it reaches a predefined threshold value.
  - *Risk*: Low
- Fixed an issue where the fault LED remains lit during the rebuild process of a replaced SSD drive.
  - *Root cause*: When a replaced SSD drive supports Over Provisioning Optimization (OPO), the firmware internally marks the drive state as bad during the Rapid Parity Initialization (RPI) to prevent any other I/O operations on this drive. However, firmware does not update the LED status. During this phase, any other configuration changes on the controller, such as logical drive creation, drive replacements, or logical drive state updates, can trigger an LED update. This causes the fault LED to remain lit throughout the rebuild process, even though the drive state has been updated to replacement.
    Fix: Firmware will skip the LED update if the drive is undergoing OPO. Additionally, a logical drive state update has been included after the completion of OPO to ensure the correct LED status is displayed.
  - *Risk*: Low

- Fixed an issue where the host was unable to retrieve new configuration updates after the creation of a logical drive.
  - *Root cause*: During the process of creating a logical drive, the firmware attempts to disable IOBypass for all physical drives involved in the request. To achieve this, the firmware waits for any outstanding IO commands on these drives to be processed. However, due to the slow processing of pending commands from a predictive failure (PF) drive, this operation took longer than expected. Consequently, new device rescans from the host experienced failures after the logical drive creation, preventing the host from recognizing new configuration updates.
  - *Fix*: Firmware will disable IOBypass before creating the logical drive. If any IO operations take an extended period of time the logical drive creation request will be aborted causing the logical drive to not be created. The logical drive creation will need to be retried.
  - *Risk*: Low
- Fixed an issue where logical volume status was showing as OK on a secured volume if a data drive is replaced with an Otherwise Owned SED drive before reboot.
  - *Root cause*: If an Otherwise Owned SED drive is used as a replacement drive on a secured volume, firmware will not fail it, which causes the logical volume state being OK.
  - *Fix*: Fail the Otherwise Owned SED with certain failure code when it's used as a replacement drive.
  - *Risk*: Low
- Fixed an issue where fault LED was not blinking for controller-based erase operation.
  - *Root cause*: On a controller-based erase operation, the firmware code controlling the LED checks the wrong value.
  - *Fix*: To ensure the fault LED blinks consistently throughout the erase operation and until its completion, the firmware will check the right variable.
  - *Risk*: Low
- Fixed an issue where long drive self-test invoked from host diagnostic tools gets aborted on inactive hot spare when spare spin-down policy is enabled.
  - *Root cause*: When long drive self-test is initiated from host diagnostic tools, firmware is not aware of drive self-test being in progress and continues with spinning down the inactive spare if spare spin-down policy is enabled resulting in drive self-test getting aborted.
  - *Fix*: Inactive spare is not spun down if drive reports self-test is running.
  - *Risk*: Low
- Fixed an issue where the surface scan period was not reset to default after factory reset.
  - *Root cause*: When a factory reset is performed, the firmware resets the persistent surface scan period variable to its default setting but fails to update the backup variable used by the host during runtime. Consequently, earlier data is displayed even after the factory reset.
  - *Fix*: The firmware will reset the local backup values of persistent data during a factory reset.
  - *Risk*: Low
- Fixed an issue where the predictive failure drive did not fail even after the Predictive Spare Rebuild (PSR) was completed.
  - *Root cause*: When a predictive failure drive is detected in an array containing a failed RAID0 logical drive, the firmware will perform a Predictive Spare Rebuild (PSR) on other operational or degraded logical drives, ignoring the failed RAID0 logical drive. The current firmware logic waits for PSR to complete on all logical drives in the array before marking the predictive failure drive as failed. Since the RAID0 logical drive is already failed and does not undergo PSR, the firmware will never mark the predictive failure drive as failed.

  - *Fix*: Updated firmware to skip waiting for PSR on failed logical drives and immediately mark predictive failure drives as failed.
  - *Risk*: Low
- Fixed an issue where the controller has a lockup after a power cycle to JBOD.
  - *Root cause*: During the power cycle of the JBOD, the firmware issued management commands and waited indefinitely, causing a deadlock.
  - *Fix*: The firmware has been updated to eliminate indefinite waiting on management commands to a JBOD.
  - *Risk*: Low
- Fixed LUN reset statistic message when the logical drive is created on a single drive.
  - *Root cause*: When a logical drive is created on a single drive, the average Queue Depth (QD) remains the same as that of a single drive. Consequently, the firmware omits printing drive details since the QD does not exceed the average.
  - *Fix*: Firmware will print the Queue Depth (QD) information for a drive that is part of a volume containing only one drive.
  - *Risk*: Low
- Fixed an issue where the offline replacement of a failed drive during a spare rebuild led to the failure of the replacement drive.
  - *Root cause*: After the offline replacement of the failed drive and upon the next boot, the rebuild process on the spare drive will be halted, and the rebuild for the replacement drive will commence. A missing check in the firmware caused the replacement drive to be incorrectly identified as a predictive failure, leading to its failure.
  - *Fix*: Implemented enhanced checks for predictive failure indicators and Predictive Spare Rebuild (PSR) within the firmware. This ensures that a drive will only fail if it is identified as a predictive failure drive and PSR is activated.
  - *Risk*: Low
- Fixed a possible lockup that could occur when a transformation is in progress and an abrupt shutdown occurs. A possible lockup could occur on subsequent bootup.
  - *Root cause*: After cold boot, the logic was checking to make sure memory is allocated properly and ended up not allocating any memory due to using a previous value from before the shutdown.
  - *Fix*: Corrected logic to make sure memory is allocated properly on new boot.
  - *Risk*: Low
- Fixed a potential controller lockup when deleting a volume with cache enabled and then changing connector mode to HBA.
  - *Root cause*: Logic was to expect volume data even though it has been deleted.
  - *Fix*: Added check to make sure if volume is present or not.
  - *Risk*: Low
- Fixed an issue where SATA Drive removed within 10 seconds after link reset due to internal firmware timer.
  - *Root cause*: An internal firmware timer value based on which a drive under reset will be immune from other PHY or port related activities is not inline with the reset wait timer, which will monitor the link reset response.
  - *Fix*: Increased internal firmware timer to 45 seconds for SATA drives, which is same duration as link reset wait timer value.
  - *Risk*: Medium

- Fixed an issue where a good drive could be labelled as a predictive fail drive incorrectly.
  - *Root cause*: There was a small scenario where an unconfigured drive could have old data and firmware used this data to mark the drive as a predictive fail.
  - *Fix*: Corrected logic to make sure the old data was not used.
  - *Fix* Risk: Low
- Fixed an issue where controller was reporting failed drives on a UBM backplane when drives were not present.
  - *Root cause*: The UBM FRU has different drive type support for different ports for different connector identities. The controller should not look at the ports that are for different connector identities, but it does, and it generates the wrong supported drive types and supported max link rate. This leads to the controller thinking the drive is present when it is not.
  - *Fix*: Only look at the ports for the connector identity when looking at supported drive types and supported max link rates.
  - *Fix* Risk: Low
- Fixed an issue where 4Ke NVMe volumes had worse sequential performance than equivalent 512e volumes.
  - *Root cause*: There was an error in logic that was assuming 512 block size instead of proper 4K size when determining to write to cache.
  - *Fix*: Corrected logic to detect correct 4K block size.
  - *Risk*: Low
- Fixed a controller lockup observed during IO to a degraded volume using IOBypass.
  - *Root cause*: The firmware sent an error response to the host for an IO that was using IOBypass by writing only a single byte instead of doing a four byte write that is required by the controller PCIe hardware. The single byte write operation caused a controller lockup.
  - *Fix*: The firmware will send an error response update to the host using a four byte write.
  - *Risk*: Low
- Fixed a drive firmware update failure due to drive report Sense command support.
  - *Root cause*: When host updates drive firmware through DOWNLOAD MICROCODE(0xE) command, after transferring all the microcode chunks, drive responds with Status with Sense Data Available. Controller on receiving the status from the drive sends REQUEST SENSE command to retrieve the Sense Data. Drive responds with CHECK CONDITION with Sense Data filled as 06: 3F: 01 - Microcode has been changed. Since CHECK CONDITION is set for REQUEST SENSE command, FATAL_ERROR is returned to upper layer hence firmware download to the drive is failed.
  - *Fix*: When drive reports KCQ Value 06: 3F: 01 - Microcode Has been changed for REQUEST SENSE command, send success to upper layer so that firmware downloaded can be activated.
  - *Risk*: Low
- Fixed a controller failure to boot when inserted into PCIe Gen 6 system.
  - *Root cause*: During boot, the controller waits for a handshake message with PCIe Gen 4 configuration details from the host. As the host device is PCIe Gen6 capable, the controller does not get the PCIe Gen 4 configuration details and waits indefinitely resulting in the controller not being found during system boot up.
  - *Fix*: Corrected the handshake message handling to work irrespective of the PCIe generation supported by the host.
  - *Risk*: Low

2.1.2 UEFI/Legacy BIOS Fixes

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

2.1.2.1 Fixes and Enhancements for UEFI Build 2.16.4/Legacy BIOS Build 2.16.3

This release includes the following fixes and enhancements:

- Added support to show drive location information in the driver health message if a previous controller lockup is detected that is caused due to a drive.
- Added a menu to configure the connector PCIe data rate and SAS Phy link rate.
- Fixed an issue where an active dedicated hot spare member is not represented in the logical drive information.
  - *Root cause*: There is no indication in the representation when a dedicated hot spare member becomes active.
  - *Fix*: New subtitle added to show active spare members under array information.
  - *Risk*: Low
- Fixed an issue where RAID level migration failed from RAID-0 to RAID-60.
  - *Root cause*: Incorrect parity count provided as input during the RAID migration.
  - *Fix*: Provide correct parity count for migration depending in the RAID level.
  - *Risk*: Low
- Fixed an issue where failed drive part of a logical drive is not shown on the HII disk utilities.
  - *Root cause*: Installed drive bit map is not considered while listing drives in the disk utilities menu.
  - *Fix*: List drives from the installed drive bit map along with the data from drive presence bit map.
  - *Risk*: Low.
- Fixed an issue where surface scan status field is shown for non-applicable RAID-0 logical drive.
  - *Root cause*: No validation for the applicability before showing the surface scan status field.
  - *Fix*: Hide Surface scan status field when not applicable.
  - *Risk*: Low

2.1.3 Driver Fixes

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

2.1.3.1 Linux Driver Fixes

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

2.1.3.1.1 Fixes and Enhancements for Linux Driver Build 2.1.32-035

This release includes the following fixes and enhancements:

- Fixed an issue where drives are not taken offline when the controller is offline. Drives are listing in sg_map and lsblk output after controller lockup.
  - *Root cause*: During a controller lockup, the physical and logical drives under the locked up controller are still listed at the OS level. The controller is offline, but the status of each drive is running.
  - *Fix*: When the controller is unexpectedly taken offline, show its drives as offline.
  - *Risk*: Low

2.1.3.2 Windows Driver Fixes

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

2.1.3.2.1 Fixes and Enhancements for Windows Driver Build 1016.10.0.1004

This release includes the following fixes and enhancements:

- Added support for Windows Server 2025.
- Added support to enable DMA remapping feature for Windows Server 2025. Kernel DMA Protection is a Windows security feature that protects against external peripherals from gaining unauthorized access to memory. Added a registry entry "DmaRemappingCompatible" under the SmartPQI services to declare the compatibility/support of the driver to the DMA protection feature.

2.1.3.3 FreeBSD Driver Fixes

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

2.1.3.3.1 Fixes and Enhancements for FreeBSD Driver Build 4570.0.1006

There are no known fixes for this release.

2.1.3.4 VMware Driver Fixes

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

2.1.3.4.1 Fixes and Enhancements for VMware Driver Build 4704.0.108

There are no known fixes for this release.

2.1.4 Management Software Fixes

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

2.1.4.1 maxView Storage Manager/ARCCONF Fixes

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

2.1.4.1.1 Fixes and Enhancements for maxView Storage Manager/ARCCONF Build 27147

This release includes the following fixes and enhancements for ARCCONF/maxView:

- Fixed an issue where the ARCCONF was allowing the user to expand the array by adding one more drive when the array has a RAID1 Triple logical drive, resulting in the raid level migration of RAID1 Triple to RAID6.
  - *Root cause*: The condition for the minimum required drives for array expansion when the array has RAID1 Triple was missing.
  - *Fix*: Made the changes for the ARCCONF expand operation on the array with RAID1 Triple will expect the number of drives in multiple of 3 to expand the array.
  - *Risk*: Low
- Fixed an issue where ARCCONF was not displaying the "S.M.A.R.T" and "S.M.A.R.T warning" property value correctly.
  - *Root cause*: Mapping of the "S.M.A.R.T" and "S.M.A.R.T warning" property value with the firmware provided values was not done correctly.
  - *Fix*: Value for S.M.A.R.T. mapped with "Supported" and "Not Supported" and S.M.A.R.T. warning value mapped with "Yes" or "No".
  - *Risk*: Low
- Fixed an issue where the maxView was not allowing to secure erase the 4K drives.
  - *Root cause*: maxView was blocking the secure erase operation for the 4K drive type.
  - *Fix*: Enabled the secure erase operation for the 4K drives in maxView.
  - *Risk*: Low
- Fixed an issue in ARCCONF where 'Negotiated Physical Link Rate' was incorrectly displayed instead of 'Physical Link Rate', and 'Negotiated Logical Link Rate' was shown instead of 'Logical Link Rate'.

- *Root cause*: The ARCCONF tool was incorrectly displaying the 'Negotiated Physical Link Rate' instead of the actual 'Physical Link Rate' and the 'Negotiated Logical Link Rate' instead of the 'Logical Link Rate'.
- *Fix*: Updated the property name 'Negotiated Physical Link Rate' to 'Physical Link Rate' and 'Negotiated Logical Link Rate' to 'Logical Link Rate'.
- *Risk*: Low

2.1.4.2 PLDM Fixes

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

2.1.4.2.1 Fixes and Enhancements for PLDM Release 6.45.7.0

This release includes the following fixes and enhancements:

- Added support for PLDM Type 7 (File I/O) compliance with final release versions of DMTF specifications
- Made the following changes to PLDM Base (Type 0) commands to comply with v1.2.0 of the PLDM base specification (DSP0240).
  - Implemented the PLDM Type 0 command `GetMultipartTransferSupport` to provide which Multipart Transfer commands the specified PLDM Type at the specified version are supported.
  - The PLDM Type 0 command `GetPLDMCommands` has been updated to report `GetMultipartTransferSupport` as a supported command.
- Made the following changes to PLDM Platform Monitoring and Control (Type 2) commands to comply with v1.3.0 of the PLDM Platform Monitoring and Control Specification (DSP0248).
  - The PDRType value for the File Descriptor PDR has been updated to 30 from the draft spec value of 25.
  - The `CrashDumpFile FileClassification` value for the File Descriptor PDR has been updated to 5 from the draft spec value of 4.
  - The supported state set values for the Device File State Sensors have been updated to conform to v1.2.0 of the PLDM State Set Specification (DSP0249).
  - The FatalHigh threshold value for file size Numeric Sensors is now set to the maximum file size.
- Made the following changes to PLDM File I/O (Type 7) commands to comply with v1.0.0 of the PLDM for File Transfer Specification (DSP0242).
  - Added support for the metacommand DfReadMultipartReceive (0x20). This command is not intended to be issued by File Clients; its only purpose is to allow the GetPLDMCommands command to indicate that the Type 0 command MultipartReceive may be used as a data transfer mechanism for PLDM Type 7.
  - Implemented the PLDM Type 7 command DfProperties to provide the maximum number of mediums supported and the total number of File Descriptors supported.
  - The command code for the Type 7 command DfHeartbeat has been updated to 0x03 from the draft spec value of 0x06.
  - Added the PLDM Type 7 completion codes UNABLE_TO_OPEN_FILE (0x8A) and ZEROLENGTH_NOT_ALLOWED (0x82).
  - DfClose can now respond with the PLDM base completion code ERROR_INVALID_DATA (0x02).
  - Support for the PLDM Type 7 draft spec completion code EXCLUSIVE_OWNERSHIP_REQUIRED (0x87) has been removed.
  - The value of the PLDM Type 7 completion code MAX_NUM_FDS_EXCEEDED has been updated to 0x88 from the draft spec value of 0x89.

- Added changes to long-running task support for volume deletion using Redfish.
  - All RDE DELETE requests for a volume resource will now result in the operation being carried out via a long-running task. BMCs are now expected to set both the delete_supported and events_supported bits of the `MCFeatureSupport` field in a `NegotiateRedfishParameters` request in order for a RDE DELETE request for a Volume resource to be allowed by the RDE device.
- Added changes to long-running task support for storage resource RDE action operations.
  - All RDE ACTION requests for a Storage resource will now result in the operation being carried out via a long-running task. BMCs are now expected to set both the action_supported and events_supported bits of the MCFeatureSupport field in a `NegotiateRedfishParameters` request in order for a RDE ACTION request for a Storage resource to be allowed by the RDE device.
- Added changes to long-running task support for volume creation using Redfish.
  - All RDE CREATE requests for a VolumeCollection resource will now result in the operation being carried out via a long-running task. BMCs are now expected to set the create_supported, events_supported, and BEJ v1.1 support bits of the `MCFeatureSupport` field in a `NegotiateRedfishParameters` request in order for a RDE CREATE request for a `VolumeCollection` resource to be allowed by the RDE device.
- Added support transfer of the controller Serial Output Buffer (SOB) log file through PLDM Type 7.
  - Added a contained entity to the Entity Association PDR having `entityType = 0x09 (Device File)` and `entityInstanceNumber = 2` representing the controller SOB log Device File.
  - Added a File Descriptor PDR with `FileClassification = 0x02 (SerialTxFIFO)` to provide a file identifier for the controller SOB log device file.
  - Added file size numeric sensor and device file state sensor PDRs to provide size and state information for the controller SOB log device file.
  - Updated the Type 7 command DfOpen to support handling for the `DfOpenRegFIFO` bit of the `DfOpenAttributes` field. When sending DfOpen for a device file that requires transmission as streaming FIFO, not setting this bit will result in a `INVALID_DF_ATTRIBUTE` error completion code in the response.
  - Updated `MultipartReceive` for type 7 to support files classified as SerialTxFIFO. The following rules and requirements apply when issuing a `MultipartReceive` request on a SerialTxFIFO file:
    - Seeking is not supported. `MultipartReceive RequestedSectionOffset` shall be set to zero.
    - Single part per section. `TransferOperation` shall not be set to `XFER_NEXT_PART`.
    - A `MultipartReceive` response where the data length is less than the negotiated part size indicates that all the available data has been transferred.
    - The response to a `MultipartReceive` request for an empty SerialTxFIFO file will have a SUCCESS completion code.
    - A `MultipartReceive` request restarting a section of a FIFO file that has wrapped will result in new data. Data overwritten by new wrapped data will not be preserved.

    The following error completion codes will be returned by `MultipartReceive` for FIFO files:
    - INVALID_DATA_TRANSFER_HANDLE if request `DataTransferHandle` is not ZERO
    - INVALID_REQUESTED_SECTION_OFFSET if request `RequestedSectionOffset` is not ZERO

    - INVALID_DATA if request `RequestedSectionLengthBytes` is ZERO or greater than the negotiated size

- Fixed an issue which threw wrong error code, '`InternalError`' when creating an array on a controller that is waiting on adapter password.
  - *Root cause*: Code does not check the granularity of the error type and returns `InternalError` as the generic error code.
  - *Fix*: Modified the following commands to return `ControllerPasswordRequired` error code when waiting on the adapter password:
    - Create Volume
    - Update Volume
    - Delete Volume
    - Update Drive
    - Update Storage
    - Update Storage Controller
  - *Risk*: Low
- Fixed an issue where the GetPDR command can sometimes fail to retrieve the requested PDR when no drives are connected to the targeted controller.
  - *Root cause*: RedfishAction PDRs for Drive resources were being internally allocated in error when no drives were present, causing a failure when an MC attempted to fetch the PDR with the GetPDR command.
  - *Fix*: Modified the logic for allocating RedfishAction PDR(s) for Drive resources to require at least one drive to be connected prior to the allocation.
  - *Risk*: Low
- Fixed an issue where with a given a configuration ExternalKey encryption is enabled and SED is controller owned and KMS is unavailable. RDE READ on the controller SED publishes Status.State and Status.Health as Enabled and OK respectively.
  - *Root cause*: The logic that sets a Drive's State and Health did not account for the case when KMS is unavailable or inactive and the Drive is a controller owned SED.
  - *Fix*: Added logic so that an RDE READ on a controller owned SED will publish Status.State and Status.Health as StandByOffline and Warning respectively when KMS is not available or inactive.
  - *Risk*: Low
- Fixed an issue where the Links.Enclosures@odata.count and Links.Enclosures@odata.id properties were missing from the Redfish storage resource.
  - *Root cause*: The Links.Enclosures@odata.count and Links.Enclosures@odata.id properties were not added to the Redfish Storage resource.
  - *Fix*: The Links.Enclosures@odata.count and Links.Enclosures@odata.id properties have been added to the Redfish Storage resource. Links.Enclosures@odata.count will contain the number of chassis resources of type enclosure being managed by the controller. Links.Enclosures@odata.id will an contain links to chassis resources of type enclosure.
  - *Risk*: Low.
- Fixed an issue for which the GetPDR for the File Descriptor PDR representing the controller crash dump did not have the Polled bit of the file capabilities field set as required by the Type 2 spec.
  - *Root cause*: The implementation of the File Descriptor PDR was based on a pre-release draft of the most recent version of the Type 2 spec, and the requirements for setting the file capabilities bits were changed during subsequent development of the spec.

  - *Fix*: Updated GetPDR to set the Polled access bit in the File Capabilities field for the controller crash dump File Descriptor PDR.
  - *Risk*: Low
- Fixed an issue in which DfOpen returns EXCLUSIVE_OWNERSHIP_NOT_AVAILABLE when opening crash dump file while it is already opened.
  - *Root cause*: Logic exists to capture this error case, but the incorrect response completion code was assigned to be returned.
  - *Fix*: Updated the error logic to send the correct completion code MAX_NUM_FDS_EXCEEDED as defined in the Type 7 spec.
  - *Risk*: Low
- Fixed an issue in which after reading the LAST_PART of a section in a file, the NEXT_PART command to read the initial part of the section must not get executed. Instead, the initial part of the section is sent and received.
  - *Root cause*: There was no check to determine if the transfer of a section was completed.
  - *Fix*: Handle the situation where the file client requests the NEXT_PART after the section has been transferred.
  - *Risk*: Low
- Fixed an issue in which the @odata.id property retrieved from an RDE READ on a drive resource representing an empty bay in a chassis resource did not match the URI given in the Chassis child drive PDR.
  - *Root cause*: The logic to generate the empty bay drive resource @odata.id property was incorrect.
  - *Fix*: The empty bay resource @odata.id property is now being calculated using the correct logic.
  - *Risk*: Low
- Fixed an issue in which incorrect severity warning was reported for the drive when sanitize erase operation is in progress on drive. The severity should be OK, and therefore the condition should not be shown.
  - *Root cause*: The severity for the offline condition was using the drive health. There was no check to determine if the drive was being erased.
  - *Fix*: When determining whether a condition should be shown for an offline drive, the check has been updated to match the check performed for an event. An offline condition will only be shown when the drive health is Warning, and the drive is not in a predictive failure state. So, when the drive is being erased and is in a predictive failure state, the offline condition will not be shown.
  - *Risk*: Low
- Fixed an issue where HealthRollup shows a warning when there is no warning on a lower level component.
  - *Root cause*: `Storage.Status.HealthRollup` was not factoring in `StorageController.Status.Health`.
  - *Fix*: Update `Storage.Status.HealthRollup` to factor in `StorageController.Status.Health`.
  - *Risk*: Low
- Fixed an issue which results in PLDM processing a `GetResourceETag` request during a RDE DELETE Volume resource operation (while deleting units through PLDM) with background drive IO would sometimes result in a controller lockup code 0xFFFFF001.

  - *Root cause*: Both the `GetResourceETag` and the Volume delete operations use a shared structure. Processing both commands results in a race condition where the thread processing `GetResourceETag` is trying to access a pointer that has been set to null by the thread processing the RDE DELETE operation. The `GetResourceETag` logic was missing a null pointer check.
  - *Fix*: `GetResourceETag` requests will now be blocked and return a completion code of NOT_READY when a long running task is ongoing. A null pointer check has been added to the `GetResourceETag` logic.
  - *Risk*: Medium
- Fixed an issue in which PLDM/RDE Read sometimes take longer than 6s while array deletion initiated with host tool.
  - *Root cause*: These RDE Reads that were timing out relied on re-initializing the cached controller information in order to complete the Read. This was fairly resource expensive and time consuming and could sometimes lead to timeouts.
  - *Fix*: Modified the logic in the RDE Reads mentioned above to not re-initialize the cached controller information.
  - *Risk*: Low
- Fixed an issue in which Incorrect CacheSummary.Status.Health Critical is observed when WriteCache is degraded. RDE READ on a StorageController was publishing the WriteCacheDegraded CacheSummary.Status.Conditions.Severity Property to be Warning, when the associated WriteCacheDegraded alert had a severity of Critical.
  - *Root cause*: All WriteCacheDegraded status conditions with a permanently disabled cache were assumed to have a CacheSummary.Status.Conditions.Severity of Warning.
  - *Fix*: Modified logic for RDE READ on a StorageController so that WriteCacheDegraded status conditions with a permanently disabled cache are now able to have a CacheSummary.Status.Conditions.Severity of Critical.
  - *Risk*: Low
- Fixed an issue to patch the volume with an Unprotected Write Cache policy on a system that supports cache without a battery has failed. Failure to set WriteCachePolicy to UnprotectedWriteBack on controllers that support caching without requiring a backup power source.
  - *Root cause*: In a previous fix, there was a logic change to set the cache ratios to the controllers default. Volume PATCH and CREATE operations do not allow setting the WriteCachePolicy to some value other than Off when the controllers default cache ratio is 100/0.
  - *Fix*: Corrected the logic to set the cache ratios to the typical default cache ratio of 10/90 when the requested WriteCachePolicy is "UnprotectedWriteBack" for both Volume CREATE and PATCH operations on controllers with default read cache of 100/0 and returning PropertyValueIncorrect when the requested WriteCachePolicy is "ProtectedWriteBack".
  - *Risk*: Low
- Fixed an issue in which A split mirror backup volume whose Status.State is StandbyOffline is showing a Volume.Status.Condition whose severity is Ok.
  - *Root cause*: The logic which determines whether a condition should be shown for a volume in an offline state was not taking the volume health into account.
  - *Fix*: Change the logic to only show a condition for a volume in an offline state when the volume health is not OK.
  - *Risk*: Low

2.2 Limitations

This section shows the limitations for this release.

2.2.1 General Limitations

This release includes the following general limitations.

- The following are the limitations of Multi-Actuator:
  - Supports only:
    - HBA drive
    - Windows/Linux/VMware
    - Intel/AMD
    - UEFI mode (for multi-LUN display)
- The NCQ Priority feature is currently not supported in this release.

2.2.2 Firmware Limitations

This section shows the firmware limitations for this release.

2.2.2.1 Limitations for Firmware Release 3.01.33.44

This release includes the following firmware limitations:

- If a boot volume is secured by Managed SED Remote Key Management (RKM) or Managed SED Adapter Password enabled Local Key Management (LKM), it will fail to write Windows memory dump file during Windows OS crash dump.
  - *Workaround*: Do not use secured volumes as described above as an OS boot logical drive.
- Persistent Event Logs (PEL) are getting cleared when:
  - Upgrading from firmware releases prior to 03.01.17.56 to 03.01.17.56 or later firmware releases.
  - Downgrading from firmware releases 03.01.17.56 or later to firmware releases prior to 03.01.17.56.
- Firmware downgrade is blocked if disk-based transformation is in-progress.
  - *Workaround*: Wait for the transformation to complete and retry the firmware downgrade.
- Transformation is blocked if a reboot is done after the firmware update is pending, and the flashed new firmware version is older than 03.01.17.56.
  - *Workaround*: Reboot the system.
- Logical drive is not detected when disk-based transformation is in-progress during logical drive movement to a different controller and the different controller has a firmware version older than 03.01.17.56, or, the firmware downgrade occurred while internal-cache based transformation was in progress, but the Backup Power Source failed before firmware activation.
  - *Workaround*: Move the logical drive to a controller with firmware version 03.01.17.56 or later.
- Firmware downgrade from firmware version 3.01.30.106 to any older firmware version is blocked if Managed SED is enabled.
  - *Workaround*: Disable Managed SED and try firmware downgrade.
- Managed SED cannot be enabled on the controller when reboot is pending after firmware downgrade from firmware version 3.01.23.72 to any older firmware version.
  - *Workaround*: Reboot the controller and enable the Managed SED.
- Flashing from 3.01.30.106 back to 3.01.28.82 or 3.01.26.36 may result in the spin down spare policy being changed to the default setting specified in the board configuration file.

- *Workaround*: If the default board configuration file specified setting is not the required setting, then re-apply the spin down spare policy using host management software.

2.2.3 UEFI/Legacy BIOS Limitations

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

2.2.3.1 Limitations for UEFI Build 2.16.4/Legacy BIOS Build 2.16.3

There are no known limitations for this release.

2.2.4 Driver Limitations

This section shows the driver limitations for this release.

2.2.4.1 Linux Driver Limitations

This section shows the Linux driver limitations for this release.

2.2.4.1.1 Limitations for Linux Driver Build 2.1.32-035

This release includes the following limitations:

- SL-Micro 6.0 fails to boot after installation on 4Kn drives.
  - *Workaround*: This is a SUSE issue and only workaround is to use non-4Kn drives.
- On some distributions (RHEL7.9, RHEL8.2, RHEL8.3, SLES15SP2, SLES15SP3, OpenEuler 20.03LTS, and 22.03LTS including SP releases), the driver injection (DUD) install will hang if an attached drive (either HBA mode or Logical Volume) has Write Cache enabled.
  - *Workaround*: There are two workarounds for this issue:
    - Ensure that the Write Cache is disabled for any attached drive.
    - For RHEL7.9/8.2/8.3 and OpenEuler 20.03LTS, 22.03LTS, add `rd.driver.blacklist=smartpqi` to the grub entry along with `inst.dd`.
- Unable to do a driver injection (DUD) install on RHEL 8.7 when NVMe drives are attached to the system. This is a multipath issue with the OS install process.
  - *Workaround*: Edit grub to include the boot argument "nompath". So replace "inst.dd" with "nompath inst.dd" for DUD install.
- RHEL driver injection (DUD) install where OS ISO is mounted as virtual media on BMC based servers (non-ILO). Installer will hang after driver injection. It is reported on RHEL 8.5, 8.6, 9.0 to 9.4.
  - *Workaround*:
    - Load the OS from USB device instead of virtual media.
    - Load the OS from virtual media but initiate ISO verification (media test) during the installation followed by ESC to cancel the media test.
    - Edit grub to include the boot argument "`nompath`". Replace "`inst.dd`" with "`nompath inst.dd`" for DUD install.
- Oracle 9 UEK 7 kernel causes SmartPQI rpm dependency failures. This is an issue with how the kernel package was created by Oracle. Correct UEK7 kernel for Oracle 9, which is expected in the mid-October UEK7 release, version number is still pending.
  Note: This does not affect Oracle 8 UEK 7.
  - *Workaround*: Install the rpm using "`--nodeps`" when dependency failures occur.
    - Update:
      For SmartPQI driver versions > 2.1.20-020 and UEK7 kernels >= 5.15.0-3.60.2.el9uek.x86_64, the SmartPQI rpm will install normally.

      For UEK7 kernels < 5.15.0-3.60.2.el9uek.x86_64, the SmartPQI rpm needs to be installed using the "`--nodeps`".

- On AMD systems, the system might crash or hang due to a bug in the IOMMU module. For details, see 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
- When multiple controllers are in a system, udev(systemd) can timeout during kdump/kexec resulting in an incomplete kdump operation. The usual indication of the timeout is the console log entry: "scsi_hostX: error handler thread failed to spawn, error = -4".
  - *Workaround*: The workaroud for this issue involves extending the udev(systemd) timeout during a kdump operation.
    - The steps to increase the timeout for udev(systemd) are:
      1. vi /etc/sysconfig/kdump
      2. Add udev.event-timeout=300 to KDUMP_COMMANDLINE_APPEND
      3. systemctl restart kdump
      4. systemctl status kdump
- On some distributions (including RHEL 9.0/Oracle Linux 9.0), you are unable to inject the OOB driver (DUD) during install when a multi-actuator drive is attached.
  - *Workaround*: Install using the inbox driver, complete OS installation, then install the OOB driver.

2.2.4.2 Windows Driver Limitations

This section shows the Windows driver limitations for this release.

2.2.4.2.1 Limitations for Windows Driver Build 1016.10.0.1004

This release includes the following limitations:

- The Windows driver issues an internal flush cache command for flushing the controller cache to the drives before changing the power state of the system (during shutdown/reboot/hibernate). Due to many factors, for example speed of drives, size of cache, type of data in cache, and so on, the time taken by the controller to flush the cached data can exceed the operating system specified timeout values. A system crash can be expected in those scenarios. Controller cache flushing will continue and complete while the system is in the BSOD state. In general, it is advised not to do heavy write operations on logical drives composed of slow drives while initiating a system shutdown in Windows 10 environments.
- A system crash may occur when hibernating a system installed on a Dual Actuator drive.
  - *Workaround*:
    - Avoid hibernating the system while running heavy I/Os to multiple Dual Actuator drives.
    - Stop running the I/Os to the drives and then hibernate the system.
    - Reboot the server to recover the system.
- A crash dump file will not be created if the system is configured with the OS system files loaded on a partition which is NOT the first partition. If the first partition is deleted and then the system happens to bug check, the crash dump file will not be written out. For example:
  1. Disk 0 is Array A
  2. Disk 1 is Array B with the OS on it
  3. If Array A is deleted and a crash dump occurs without a reboot, the OS will NOT write out the crash dump file.

  – *Workaround*: This is only seen in the above configuration and if the deletion is done without doing a system reboot. To avoid the problem, make sure the OS is on the first partition or ensure that any time an array is deleted the system is rebooted.

- A Logical drive goes into an offline state after a new array migration.
  - *Workaround*:
    i. Perform logical disk migration.
    ii. Run DiskPart.
    iii. Run the command "List Disk" to identify all the physical disks that have a duplicate unique disk IDs.
    iv. Run the command "Select Disk X", where X is the physical disk with the duplicate Unique disk ID to be cleaned.
    v. Run the command "clean". This cleans the physical disk with the duplicate disk ID (aka partition ID).
    vi. Run command "select disk Y" where Y is the newly migrated logical disk.
    vii. Run the command "online disk", which will bring the migrated logical drive online.

2.2.4.3 FreeBSD Driver Limitations

This section shows FreeBSD driver limitations for this release.

2.2.4.3.1 Limitations for FreeBSD Driver Build 4570.0.1006

This release includes the following limitation:

- FreeBSD 13.2 and later OS Installations will fail with the out of box driver.
  - *Workaround*: Install with inbox driver then update to latest.

2.2.4.4 VMware Driver Limitations

This section shows VMware driver limitations for this release.

2.2.4.4.1 Limitations for VMware Driver Build 4704.0.108

This release includes the following limitations:

- If the controller SED Encryption feature is "On" and locked, Datastores created from secured logical drives on the controller are not automatically mounted even after unlocking the controller, they are not visible through the ESXi hypervisor client.
  - *Workaround*: Use the command `vmkfstool -V` or ESXCLI storage filesystem rescan. Alternatively, use the Rescan option from the Devices tab in the Hypervisor's Storage section. Any of these options solve the issue by forcing a rescan, causing the datastore to mount.
- A controller lockup may occur when using VMDirectPath on a single-processor AMD system. These lockups have been seen with VMs running Linux and Windows. If a lockup of a passed-through controller occurs, a reboot of the ESXi server may be required to clear the lockup condition and restore the virtual machine to working condition.
  - *Workaround*: No known workaround at this time.
- Customers may encounter failures when attempting to add new Logical Drives (LD), particularly in cases involving a dead path.
  - *Workaround*: To facilitate recovery of new LD, customers are required to clear the dead path initially. Following the clearance of the dead path, if the newly created LD is still not exposed, then it is required to initiate a driver level rescan using the appropriate management tool. If clearing the dead path fails, a host reboot is required.
- Creating a Virtual Machine File System (VMFS) datastore using a RAID 5 logical drive initialized using Rapid Parity Initialization and using IOBypass on a cache-less adapter may result in the datastore not being accessible.
  - *Workaround*: Disable IOBypass and recreate the VMFS datastore on the logical drive.

2.2.5 Management Software Limitations

This section shows management software limitations for this release.

2.2.5.1 maxView Storage Manager/ARCCONF Limitations

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

2.2.5.1.1 Limitations for maxView Storage Manager/ARCCONF Build 27147

There are no known limitations for this release.

2.2.5.2 PLDM Limitations

This section shows the PLDM limitations for this release.

2.2.5.2.1 Limitations for PLDM Release 6.45.7.0

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.

Important: When downgrading firmware, there may be cases when newer hardware is not supported by an older version of firmware. In these cases, attempting to downgrade firmware will be prevented (fail). It is recommended to regularly qualify newer firmware versions, to ensure that newer hardware is supported in your system(s)

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 3.0X.XX.XXX Firmware

1. For controllers running 3.01.02.042 or higher firmware, flash with 3.0X.XX.XXX 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
R	12/2024	Updated for SR 3.4.2 release.
Q	07/2024	Updated for SR 3.4.0 release.
P	02/2024	Updated for SR 3.3.4 release.
N	11/2023	Updated for SR 3.3.2 release.
M	10/2023	SR 3.3.0 patch release with maxView™ version B26068.
L	10/2023	SR 3.2.0 patch release with maxView™ version B25339.
K	08/2023	Updated for SR 3.3.0 release.
J	03/2023	Updated for SR 3.2.4 release.
H	11/2022	Updated for SR 3.2.2 release.
G	07/2022	Updated for SR 3.2.0 release.
F	02/2022	VMware driver version changed from 4250.0.120 to 4252.0.103.
E	02/2022	Updated for SR 3.1.8 release.
D	12/2021	Updated for SR 3.1.6.1 release. Updated Fixes and Enhancements for maxView Storage Manager/ARCCONF section for log4j vulnerabilities.
C	11/2021	Updated for SR 3.1.6 release.
B	08/2021	Updated for SR 3.1.4 release.
A	06/2021	Document created.

Microchip Information

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, 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, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, 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, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, 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, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, 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.

ISBN: 979-8-3371-0221-4

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

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.

Microchip Devices Code Protection Feature

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

- 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, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- 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. Microchip is committed to continuously improving the code protection features of our products.