HBA 1200 Software/Firmware Release Notes

Table of Contents

1. About This Release

The 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	3.4.4
Package release date	April 30, 2025
Firmware version	3.01.36.50
UEFI/Legacy BIOS	2.18.4/2.18.2
Driver versions	**Windows Drivers**: • Windows 2025, 2022, 2019, Windows 11, 10: 1016.18.0.1014 **Linux SmartPQI**: • Rocky Linux 9: 2.1.34-035 • RHEL 7/8/9: 2.1.34-035 • SLES 12/15: 2.1.34-035 • Ubuntu 20/22/24: 2.1.34-035 • Oracle Linux 7/8/9: 2.1.34-035 • Citrix Xenserver 8: 2.1.34-035 • Debian 11/12:2.1.34-035 **VMware**: • VMware ESX 7.0/8.0: 4856.0.105 **FreeBSD**: • FreeBSD 14/13: 4620.0.1010
ARCCONF/maxView	4.26.00.27449
PLDM	6.50.11.0

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, 9.4, 9.0, 8.10, 8.9, 8.0, 7.9	x86_64
SLES 12 SP5	x86_64
SLES 15 SP6, SP5	x86_64
Ubuntu 20.04.6, 24.04.1, 24.04, 22.04.5, 22.04.4, 20.04	x86_64
Ubuntu 22.04	x86_64
Oracle Linux 7.9 UEK6U3	x86_64
Oracle Linux 9.5, 9.4, 8.10, 8.9 UEK7U3	x86_64
Debian 12.8, 12.6, 11.11, 11.10	x86_64
Fedora 41 (inbox)	x86_64
Citrix XenServer 8.2.1	x86_64
Rocky Linux 9.4, 9.3	x86_64
SLE-Micro 6.1, 6.0 (inbox only)	x86_64

Table 1-6. FreeBSD and VMware Drivers

OS	Version
VMware 8.0 U3/U2, 7.0 U3/U2	x86_64
FreeBSD 14.2, 13.4	x86_64

Note: Though provided driver bundle includes drivers for several other OSes, only versions mentioned above have been QA tested and are officially supported in this release.

Host Management Software

Table 1-7. maxView^{TM} 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^{TM} 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^{TM} 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 03.01.36.50

This release includes the following fixes and enhancements:

- Added support for exposing the form factor of NVMe drive as defined in the NVMe-MI specification.
- Added support to clear cache error using PLDM.
- Added support to prevent firmware downgrade to old versions that do not support BiCS5 NAND.
- Fixed an issue where controller was failing to take ownership of some OPAL drives.
  - *Root cause*: The firmware was using a hard-coded password length instead of using length of key returned fromdrive.
  - *Fix*: Use length of key returned from drive.
  - *Risk*: Low
- Fixed an issue that otherwise owned SED changes back to foreign SED after a controller reboot
  - *Root cause*: When deleting a foreign volume, the foreign SEDs in the volume are set to otherwise owned SEDs by the firmware. The otherwise owned flag is not saved to the SED's datastore. After a controller reset, the firmware reads the SED's datastore, gets the foreign SED flag, not the otherwise owned flag.
  - *Fix*: When the firmware sets the otherwise owned flag for the SED, saves it to the datastore. When reboot happens, the firmware can read the otherwise owned flag and restore it to otherwise owned state.
  - *Risk*: Low
- Fixed a firmware crash/lockup issue when the Non-Disruptive Software Reset (NDSR) operation is performed on the expanders during the logical drive rebuilding and a drive is failed by firmware.
  - *Root cause*: During NDSR operation and the logical drive rebuilding, firmware detected the I/O timeout on the rebuilding drive and issued a device reset to this drive. After NDSR operation, firmware restarted the device discovery and reset some internal variables. At the same time right after NDSR operation, firmware failed the I/O timeout drive due to the number of write retries failure on this drive. During failing the I/O timeout drive, firmware tried to clean up all outstanding requests already delivered to this failed drive and accidentally referencing the NULL pointer variable was reset at the restart of the device discovery. This leads to firmware crash/lockup.
  - *Fix*: During clean up all outstanding requests for the failed drive, firmware will check for the NULL pointer variable and skip the cleanup process.
  - *Risk*: Low
- Fixed an issue where 4Kn SED physical drive tries to rebuild into a 512B block logical drive.
  - *Root cause*: During the device hot-plug, firmware validates the replacement drive for block/ sector size. If it is a mismatch, firmware fails the drive with the failure reason code of the non-MSED type mismatch in hot plug. But in this case the hot-plugged drive is MCHP foreign SED and is in the locked SED state, it cannot be failed with the non-MSED failure reason code. As it is not in the allowable failure reason code list for MSED support. As the result, this 4Kn

MCHP foreign SED is still exposed to the host and allows it to be imported. After the import, it leads to the volume rebuild and drive failure on this 4Kn MCHP foreign SED.
  - *Fix*: The 4Kn MCHP foreign SED should be failed with the failure reason code of the MSED type mismatch in hot plug. So, it won't be exposed to the host and allow the import. This 4Kn MCHP foreign SED must be replaced with a SED that matches the block size.
  - *Risk*: Low

- Fixed an issue where drive connector details were incorrectly logged in persistent events.
  - *Root cause*: The firmware, while logging persistent events with drive details, attempted to convert the connector name provided by the hardware. If the connector name was limited to two characters, the conversion would fail, preventing the data from being captured and posted into persistent events.
  - *Fix*: Updated the firmware to include a condition that checks the string length of the connector name. If the name is too short, it is converted to the required format to ensure proper logging in persistent events.
  - *Risk*: Low
- Fixed an issue where BMC unable to retrieve controller information due to timeout errors.
  - *Root cause*: The firmware's Out-of-Band (OOB) message threshold timeout was set to 100 milliseconds. This timeout was insufficient in scenarios where the firmware required additional processing time due to workload, resulting in commands timing out after 100 milliseconds.
  - *Fix*: The firmware's OOB message threshold timeout was increased to 300 milliseconds to ensure adequate processing time. This adjustment aligns better with the BMC's request timeout of 500 milliseconds, preventing unnecessary timeout errors.
  - *Risk*: Low
- Fixed an issue where tools and the BMC did not report a physical drive's status as Predictive Failure (PF), even when the PF LED was lit.
  - *Root cause*: Some physical drives set the DEXCPT bit to 1 but still report predictive failure warnings. In such cases, the firmware was updating the drive status and LEDs accordingly, but tools and BMC were not.
  - *Fix*: The firmware has been updated to check whether the DEXCPT bit on the physical drive is configurable. If the drive supports configurability, the firmware will reset the DEXCPT bit to 0. This ensures that Predictive Failure status is properly reflected across tools, the BMC, and the PF LED, providing consistent and accurate reporting.
  - *Risk*: Low
- Fixed an issue where firmware incorrectly reporting the queue depth for Zoned Block Devices (ZBD) during hot-plug event.
  - *Root cause*: During the handling of ZBD physical drive hot-plug, the firmware prematurely exited the setup process before updating the queue depth reported by the physical drive.
  - *Fix*: The firmware has been updated to ensure that the queue depth is correctly updated in the drive parameters of the ZBD physical drive during hot-plug handling.
  - *Risk*: Low
- Fixed an issue where controller was failing drive after hot plug event.
  - *Root cause*: UBM backplane was incorrectly reporting an installed drive as unsupported.
  - *Fix*: Put a FW workaround in to not fail an installed drive that reports unsupported and gives the backplane a chance to correct itself.
  - *Risk*: Low
- Fixed an issue where drive LED was being changed unexpectedly.

  - *Root cause*: LED command was being sent even if value read was same value to be sent.
  - *Fix*: Do not send command if value does not need to change.
  - *Risk*: Low
- Fixed an issue to remove a firmware assert for invalid frame length for incoming SSP sense data.
  - *Root cause*: A controller connected to cascaded expanders gets an SSP response IMQ with invalid frame length. There is an assert which locks up the controller with code 0x9176.
  - *Fix*: To avoid assert, added high severity log to capture invalid frame packet and sense data frame length set to 0 for further processing.
  - *Risk*: Low
- Fixed an issue where NVMe drives fail after an SEP firmware update.
  - *Root cause*: A mixed topology with SAS/SATA drives connected across MCHP 24G expander and NVMe drives connected directly to the controller through a UBM backplane. Controller was dropping the NVMe drive after expander firmware update and reset.
    As a part of topology discovery in the controller, there are device scans to add back the hot-added devices after an expander reset. This device scan triggers an event to the RAID stack to let it wait on the hot plug events to add back the drives. These NVMe hot plug events were missed by the RAID stack due to a late basecode event to the RAID stack failing to add back cleared drives.
  - *Fix*: Send PAL_EVT_SCAN_DEV_START(0x16) event before NVMe devices discovery as well as before SAS/SATA devices discovery. RS now hot adds both NVMe and SAS/SATA drives in parallel.
  - *Risk*: Medium
- Fixed an issue where the hardware was queueing new commands automatically to drive after a link reset due to uncleared residue.
  - *Root cause*: When a loss of sync occurs, the controller starts queuing NCQ commands right after receiving the D2H Init FIS from the drive, even before the firmware sends the Identify Frame to the drive.
  - *Fix*: To prevent commands from being queued to the drive, the ITC skip bit is set for the specific drive when the PHY goes down. The skip bit is reset when the device I/O unfreezes, ensuring that the hardware does not initiate any transactions with the drive during this period.
  - *Risk*: Low
- Fixed an issue where an unaligned SGL to PRP conversion for transfer length of 2M which was causing a RHEL >8.8 boot up failure.
  - *Root cause*: In two scenarios, NVMe IO read requests failed: when the read request size was 2M, and when the drive's Maximum Data Transfer Size (MDTS) was 2M. The previous fix adjusted data transfers larger than 2M with unaligned start addresses by modifying the data length in the 513th PRP page. This involved offsetting the data length with the alignment offset from the first PRP page. However, this approach did not account for 2M transfers with unaligned start addresses, leading to a PRP hole due to the offset being applied even in these cases.
  - *Fix*: Fix the data length to be considered for the current I/O before starting the PRP generation instead of handling after the PRPs are generated. The data length should be fixed to controller limitation of 2M or MDTS of drive whichever is minimum.
  - *Risk*: Medium
- Changed the drive dampen timer value for expander-attached SATA drives.
  - *Root cause*: A dampen timer is used to track the time from which the drive has been queued up in SSU queue. In the expander, based on SSU algorithm drives queued up in SSU queue

will be spun up. From controller side, a dampening timer is used to wake-up the drive after the drive in SSU queue has expired, which is not in sync with the proposed TTR time by drive vendor in case of SATA drives. Here the SATA drive is getting hot removed within 10 secs, which is not expected behavior from BC side.
  - *Fix*: During boot-up the drive dampen timer value will be initialized to 10 seconds, which is default and will be used only for SAS drives. But when the timer is getting started, if it is a SATA drive, the drive dampen timer value will be overwritten to 45 seconds, which is the TTR value proposed by SATA drive vendors.
  - *Risk*: Low
- Fixed an issue with VPD read data length and conversion of device handle.
  - *Root cause*:
    1. As a part of NVMe drive discovery, an NVMe-MI VPD Read command will be triggered in order to retrieve the drive form factor. In VPD Read, Multi Record Area contains the form factor information. From the controller side, while constructing the NVMe-MI command, data length expecting is wrong.
    2. In the normal existing code flow, the device handle shared by PAL layer will be converted to the NVMe layer device handle by NVMe SCSI layer. This would be understood by the NVMe layer. As the NVMe SCSI layer is getting skipped while sharing the form factor value with PAL layer, conversion of device handle is not handled properly.
  - *Fix*:
    1. While constructing VPD Read command, the expected data length has been changed to 256 bytes.
    2. Conversion of PAL device handle according to NVMe layer, before calling NVMe layer has been done.
  - *Risk*: Low
- Fixed corrupted VPD Read multi record from drive by Flushing L1/L2 Cache.
  - *Root cause*:
    After getting the requested VPD Read Multi Record Area information from the drive and storing in local DRAM. The first few bytes of the received data are being corrupted.
  - *Fix*: Invalidating the controller L1/L2 cache will flush the latest data to the assigned DRAM location.
  - *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 2.18.4/Legacy BIOS 2.18.2

This release includes the following fixes and enhancements:

- Added enhancement to show separate fields for the drive vendor and model name in the disk information.
  - Implementation Details: The Disk Information menu will show separate fields for the drive vendor and drive model data.
- Added support to transfer encryption keys securely in remote key management mode.
  - Implementation Details: Encryption keys received from the EFI KMS protocol, which are required for Controller Based Encryption and Controller Managed SED remote mode encryption, are transferred securely by the UEFI driver.
- Added support to enhance Drive write cache status information.

  - Implementation Details: The drive write cache status information in HII has been updated to handle the case for NVMe drives that do not support the command to retrieve the cache status.
- Fixed an issue where disk utilities display an invalid box number for drives after failing the active path.
  - *Root cause*: Incorrect box number shown for the alternative path.
  - *Fix*: Show the box number from the alternative path only if the controller marks it as valid.
  - *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 2.1.34-035

This release includes the following fixes and enhancements:

- Fixed an issue where the kernel call trace when calling `smp_processor_id()` in real-time kernel.
  - *Root cause*: `smp_processor_id()` checks to see if preemption is disabled. If enabled, it will issue an error message followed by a call to `dump_stack()`. `smp_processor_id()` can potentially return an inaccurate CPU ID if a context switch happens during its execution, while `raw_smp_processor_id()` is designed to avoid this issue by holding a lock internally while retrieving the ID. This makes it more reliable for CPU identification during highly time-sensitive situations.
  - *Fix*: Switch to using `raw_smp_processor_id()`.
  - *Risk*: Low
- Fixed a rare race condition between our scan thread and offline handler.
  - *Root cause*: The scan thread was removing a SCSI device before the offline handler could access the SCSI device pointer to set its state to OFFLINE. The offline handler has been updated to check for a NULL SCSI device pointer and the device list is now protected with a device_list lock.
  - *Fix*: Add check for null sdev in pqi_take_ctrl_devices_offline() function.
  - *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 1016.18.0.1014

There are no known fixes and enhancements for this release.

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 4620.0.1010

There are no known fixes and enhancements 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 4856.0.105

This release includes the following fixes and enhancements:

- Fixed an issue where initialization performs a capabilities inquiry to get values for max elements, and maximum and minimum size of the elements.

  - *Root cause*: PSOD indicates a divide-by-zero happened when computing a value from PQI capability structure member `max_iq_elem_len`. Driver checks for capability values that do not align with driver expectations, but only prints a warning rather than halting the initialization process.
  - *Fix*: Show error message and exit the initialization smoothly if capabilities data do not match the driver's expected values.
  - *Risk*: Low

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 27449

This release includes the following fixes and enhancements:

- Added support in ARCCONF GETSTATUS command to filter and display the task details per logical/physical device.
- Fixed an issue where the ARCCONF was not enabling the **ssdoverprovisioningoptimization** feature on the SSD logical device.
  - *Root cause*: The **ssdoverprovisioningoptimization** feature was not mapped properly which resulted to disable this feature for the SSD logical device.
  - *Fix*: The **ssdoverprovisioningoptimization** feature is mapped to the correct feature value.
  - *Risk*: Low
- Fixed an issue where the UEFI ARCCONF was not is not triggering the LED blinking on associated array hot-spares, when the Array level locate operation was executed.
  - *Root cause*: The associated array hot spare devices were not included while array level locate operation was executed.
  - *Fix*: Added the changes to blink the associated array hot-spare when identifying array operation is executed.
  - *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 6.50.11.0

This release includes the following fixes and enhancements:

- Added support that require privileged operations token for all RDE ACTION operations. Certain controllers require a custom request header to be sent to the RDE device for data destructive RDE Write operations. The current implementation only applies this condition to the Drive.SecureErase and Storage.ResetToDefaults RDE ACTIONS and all RDE CREATE, DELETE, and UPDATE operations. Modified the helper function which determines whether or not an RDE operation is data destructive to consider all RDE ACTION operations to fall in that category. This will result in the Drive.Reset and Storage.SetEncryptionKey RDE ACTIONs now requiring a custom request header to passed in with the RDE operation request.
- Added support to provide meaningful updateInterval value for Numeric Sensor PDRs.
  - Set the updateInterval to 5 seconds for the controller temperature Numeric Sensor PDR.
  - Set the updateInterval to 60 seconds for the 'hottest or normalized' drive temperature Numeric Sensor PDR.

  - Set the updateInterval to 60 seconds for individual drive temperature Numeric Sensor PDRs on controllers supporting these PDRs.
- Added support to handle failure to fetch NVMe drive WCE setting. RDE READ responses for NVMe Drive resources will now have the WriteCacheEnabled property published with type bejNull instead of the previous bejBoolean if the controller fails to fetch the WCE setting from the drive.
- Added support for the following DriveFormFactor enum values for RDE READ operations on NVMe Drive resources:
  - EDSFF
  - EDSFF_1U_Long
  - EDSFF_1U_Short
  - EDSFF_E3_Long
  - EDSFF_E3_Short
  - M2_22110
  - M2_2230
  - M2_2242
  - M2_2260
  - M2_2280
  - PCIeHalfLength
  - PCIeSlotFullLength
  - PCIeSlotLowProfile
  - U2

  Note: The DriveFormFactor property will not be published for unsupported values.
- Fixed an issue where the controller firmware was not synchronizing status between PLDM Type 2 (PDR) and Type 6 (RDE) data. The present state of the controller composite state sensor Health state can sometimes have a reading that does not match that for the Status.HealthRollup property value obtained from a RDE READ on the Storage resource.
  - *Root cause*: A previous fix to the logic used to determine the value to publish for Storage.Status.HealthRollup did not apply that same fix to the controller composite state sensor.
  - *Fix*: Modified the handling of GetStateSensorReadings requests for the controller composite state sensor to ensure that the Health state reading will match the value of Storage.Status.HealthRollup obtained from an RDE READ operation on the Storage resource.
  - *Risk*: Low
- Fixed an issue where SSD firmware update is failing due to PLDM error. RequestFirmwareData command when performing firmware update on a SAS drive fails when the drive is in a dual path configuration.
  - *Root cause*: RequestFirmwareData sends a SCSI WRITE BUFFER command with mode 0xE. If this SCSI command is sent to the inactive path of the drive, it can cause the drive to return a check condition. PLDM fails the command if a check condition is returned by the drive.
  - *Fix*: There are two changes:
    - Before performing the first SCSI WRITE BUFFER command, PLDM will now send a Test Unit Ready command to force the drive to clear any check conditions.
    - If the update agent performs a self contained activation via ACTIVATE FIRMWARE command, PLDM will send a Test Unit Ready command to clear any check conditions to ensure the drive's inactive path can accept SCSI passthrough commands.
  - *Risk*: Low

- Fixed an issue where continuous serial log prints were observed when Predictive Failure state PD is Failed in system. When a drive in predictive failure goes into failure, periodic SCSI error prints appear in the controller serial log.
  - *Root cause*: The SCSI error prints are related to repeated attempts to fetch drive monitoring and performance statistics in response to a RDE READ request incoming for the drive's associated DriveMetrics resource. A bug recently introduced into the API requesting this data allowed the request to be sent to controller firmware in spite of the failed status of the drive.
  - *Fix*: Updated the API submodule to pick up a fix for the erroneous behavior of the API.
  - *Risk*: Low
- Fixed the incorrect OriginOfCondition in Redfish Message Events when battery is in failure state. Battery Redfish Alerts are generated with incorrect OriginOfCondition data. The OriginOfCondition field contains a link reference to a Redfish Storage Controller resource, rather than a Battery resource.
  - *Root cause*: The OriginOfCondition field in the Battery Event is being populated with a StorageController URI. The OriginOfCondition field should contain a link reference to the Redfish resource associated with the Redfish event.
  - *Fix*: Modified the OriginOfCondition URI for Battery Redfish Alerts to set a link to the Battery resource.
  - *Risk*: Low
- Fixed an issue where performing an RDE UPDATE operation on a Volume resource's ReadCachePolicy property was completing successfully while a CBE rekey was running on the Volume. However, a follow-up RDE READ on the Volume showed that no change to the Volume's ReadCachePolicy was made.
  - *Root cause*: Much of the cache RDE UPDATE logic is based on the controller cache status being either unconfigured or okay. In the case of this issue, the controller cache status was temporarily disabled, leading to unexpected behavior in the cache RDE UPDATE logic. Although there were checks already in place to block the cache patch if the controller cache status is temporarily disabled, these checks were not being hit when the controller cache status was in that state.
  - *Fix*: Adjusted the cache RDE UPDATE logic to no longer depend on the cache status being unconfigured or okay. Accurate error messages will now be returned either from logic within the cache patching function or internal API's called by the function, regardless of controller cache status.
  - *Risk*: Low
- Fixed an issue where RDE READ for a Drive resource shows the Status.State is enabled for drives which are ATA security locked and are blocking all IO requests.
  - *Root cause*: PLDM had no means of detecting the ATA security locked state for drives and thus had no logic in place to handle this test case.
  - *Fix*: Picked up API changes which expose the ATA security enabled / locked states for a drive. Updated the helper functions which determine drive state for both RDE READ and event generation purposes. RDE READ for a Drive resource will now have Status.State of StandbyOffline and Health of Ok for ATA security locked drives. A DriveOffline Redfish alert event will be generated at boot when a drive is powered on in the ATA security locked state, and a DriveOfflineCleared alert will be generated at some point after the ATA security password has been supplied. Note that the controller firmware does not support configuration of drives with ATA security enabled, i.e. with a password set, regardless of locked status.
  - *Risk*: Low
- Fixed an issue where the HotspareActivationPolicy was showing OnDriveFailure in an allowable value when configure only RAID-0 with Dedicated spare. RDE READ on the Storage resource

shows "OnDriveFailure" in the HotspareActivationPolicy@Redfish.AllowableValues array when volume configuration constraints would make this setting invalid, for example, having a RAID 0 Volume with a dedicated spare.

  - *Root cause*: The logic for populating the HotspareActivationPolicy@Redfish.AllowableValues array was only checking for controller support and not inspecting the current configuration of Volumes.
  - *Fix*: Added checks of Volume configuration to make a more accurate determination of when to add the "OnDriveFailure" value to the HotspareActivationPolicy@Redfish.AllowableValues array.
  - *Risk*: Low

- Fixed an issue where there is a mismatch in Overall Health State and Logical Volume State for PLDM_OTHER_STORAGE_DEVICE_ENTITY_TYPE during RPI. While a Volume is undergoing RPI, GetStateSensorReadings for the logical drive state sensor would show the presentState as Warning, but the Status.Health property for all Volume resources would remain at OK when inspected via a RDE READ operation.
  - *Root cause*: Logic which sets Volume Status.Health to Ok while RPI is underway was not being applied when calculating the presentState of the logical drive state sensor.
  - *Fix*: The logic to process a GetStateSensorReadings request on the logical drive state sensor has been updated to prevent RPI from resulting in a presentState of Warning.
  - *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)

2.2.2. Firmware Limitations

This section shows the firmware limitations for this release.

2.2.2.1. Limitations for Firmware Release 03.01.36.50

This release includes the following 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 with an OS boot logical drive.
- Persistent Event Logs (PEL) will be cleared under the following conditions:
  - 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.
- There is a chance of controller lockup if configuration changes (set_config commands) are issued simultaneously from the out-of-band (OOB) management path during high I/O workload testing. While this scenario does not result in data loss, it can cause temporary disruption to system operations.
  - *Workaround*: Reboot the system.

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 2.18.4/Legacy BIOS Build 2.18.2

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 2.1.34-035

This release includes the following limitations:

- A call trace may be observed when performing a drive hot removal/re-add while I/O is running. This is seen exclusively on RHEL9.4 and has been tracked down to a kernel bug in the blk-mq subsytem. It has been patched starting with `kernel.org kernels 6.14-rc1`.
  - *Workaround*:
    i. Stop I/O before doing drive hot removals/additions.
    ii. Use a non-affected Linux OS release.
    iii. Update to a later version of RHEL9.
- OS installation hangs when attempting to load the OOB SmartPQI driver DUD.
  - *Workaround*: This failure occurs when using `inst.dd` even if no driver update is provided. This is an OpenEuler 24.03 installer issue.

- 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.
  - *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.
- 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.
- 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*: There is a workaround for this issue which involves extending the udev(systemd) timeout during a kdump operation. The steps to increase the timeout for udev(systemd) are as follows:

    i. vi /etc/sysconfig/kdump

    ii. add udev.event-timeout=300 to KDUMP_COMMANDLINE_APPEND

iii. systemctl restart kdump

iv. systemctl status kdump

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:

- 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:

  a. Disk 0 is Array A

  b. Disk 1 is Array B with the OS on it

  c. 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 4620.0.1010

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 4856.0.105

This release includes the following limitations:

- A system may PSOD if attached JBOD enclosures are power cycled while the system is running.
  - *Workaround*: Avoid powering OFF JBOD enclosures while the system is running.
- 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.

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 27449

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 6.50.11.0

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 or security/software features are not supported by an older version of the firmware. In these cases, attempting to downgrade the firmware will be prevented (fail). It is recommended to regularly qualify newer firmware versions to ensure proper support 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
S	04/2025	Updated for SR 3.4.4 release.
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	SR 3.3.0 patch release with maxView™ version B26068.
M	11/2023	SR 3.2.0 patch release with maxView™ version B25339.
L	11/2023	Updated for SR 3.3.2 release.
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: 979-8-3371-1151-3

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.