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?....6
   - 2.1. Fixes and Enhancements....6
   - 2.2. Limitations....16
3. Updating the Controller Firmware....19
   - 3.1. Updating Controllers to Latest Firmware....19
4. Revision History....20

The Microchip Website....21

Product Change Notification Service....21

Customer Support....21

Microchip Devices Code Protection Feature....21

Legal Notice....21

Trademarks....22

Quality Management System....23

Worldwide Sales and Service....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	3.2.2
Package release date	November 28, 2022
Firmware version	3.01.17.56
UEFI/Legacy BIOS	2.4.1/2.4.3
Driver versions	**Windows Drivers**: • Windows 2022, 2019, 2016, Windows 11, 10: 1010.52.0.1012 **Linux SmartPQI**: • RHEL 7/8/9: 2.1.20-035 • SLES 12/15: 2.1.20-035 • Ubuntu 18/20/21/22: 2.1.20-035 • Oracle Linux 7/8/9: 2.1.20-035 • Citrix Xenserver 8: 2.1.20-035 • Debian 10/11: 2.1.20-035 **VMware**: • VMware ESX 7.0/8.0: 4380.0.108 **FreeBSD/Solaris**: • FreeBSD 12/13: 4330.0.1038 • Solaris: 11: 11.4120.0.1005
ARCCONF/maxView	4.09.00.25611
PLDM	6.15.13.0

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 2022, 2019, 2016, Windows 11, 10	x64

Table 1-5. Linux Drivers

OS	Version
RHEL 9.1[1], 9.0[2], 8.7[1], 8.6, 8.5, 8.4, 7.9, 7.8	x64
SLES 12 SP5, SP4	x64
SLES 15 SP4, SP3, SP2	x64
Ubuntu 20.04.5, 20.04.4, 20.04, 18.04.5, 18.04.4	x64
Ubuntu 22.04.1, 22.04, 21.04	x64
Oracle Linux 7.9 UEK6U3	x64
Oracle Linux 9.0, 8.6 UEK7	x64
Debian 11.4, 10.12, 10.10	x64
Fedora 36 (inbox)	x64
Citrix XenServer 8.2.1	x64

Notes:

1. New OS support—minimally tested drivers in this release. Fully supported drivers are expected in the next release.
2. Support based off August 2022 RHEL 9.0 ISO refresh.

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

OS	Version
ESX 8.0, 7.0 U3/U2	x64
FreeBSD 13.1, 12.3	x64
Solaris 11.4	x64

Host Management Software

Table 1-7. maxView™ and ARCCONF Utilities

Description	OS	Executable
ARCCONF Command Line Utility	Windows x64 Linux x64 VMware 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 03.01.17.56

This release includes the following fixes and enhancements.

- Added support for a new persistent event log policy that overwrites old events with the most recently occurred events.
- Added support for Managed SED (MSED).
- Added a feature to prevent downgrading to prior firmware release that do not support MSED if MSED is enabled.
- Revert workaround that downshifts PHYs to a maximum rate of 12 Gbps when SATA drive is expander attached.
- Disabled Dynamic Channel Multiplexing (DCM) when all targets are 24G.
- Added SPDM certificate storage and management commands.
- Logical drive deletion when backup power source is not present.
- Enabled UEFI KMS support for controller-based encryption.
- Added a workaround for a SATA spin-up hold issue observed when communicating with a Broadcom expander.
  - *Root cause*: When a SATA drive behind a Broadcom expander is in spin-up hold, the expander does not set the logical link rate in SMP DISCOVER to SPINUP_HOLD, but rather sets the Attached SATA drive bit and sets the logical link rate to unknown. This causes logical drives to enter a Fail or Rebuild state, upon power cycle due to physical drives missing from initial discovery.
  - *Fix*: If an SMP discover response indicates attached SATA drive and an unknown logical link rate, firmware treats it as a spin-up hold case.
  - *Risk*: Low
- Fixed a parity inconsistency after RAID rebuild restarted with a new drive replacement.
  - *Root cause*: Every 10 minutes during the rebuild process or when a graceful system shutdown occurs, firmware will save the rebuild progress information to RAID metadata. If a rebuilding drive is replaced with a new drive, the firmware will restart the rebuild from the beginning of the drive. If the system has an ungraceful shutdown within the first 10 minutes of restarting the rebuild, the rebuild will continue from the previously saved rebuild progress information in RAID metadata and run to completion, which will result in skipped blocks during the rebuild leaving incorrect data or parity in the rebuilt drive. When consistency check runs, it detects inconsistent parity on the skipped blocks and fixes them. The file system or the application can end up reading the incorrect data from those skipped blocks.
  - *Fix*: Firmware saves the Rebuild Progress Information in RAID metadata at the start of a rebuild.
  - *Risk*: Low
- Fixed an issue where the drive fails during the drive firmware update.
  - *Root cause*: After the drive firmware update, the host re-enquires about the drive parameters. The firmware will reset the existing drive parameters information and fill up the data again by re-enquiring about the drive. If there is a simultaneous firmware operation accessing drive parameters, it ends up with wrong values and fails. After a defined number of retries on firmware operation, the firmware fails the drive.
  - *Fix*: Avoid resetting the default drive parameter value fields for host drive parameter during re-inquiry.
  - *Risk*: Low
- Fixed an issue where OS possibly places logical drives containing multiple predictive failed physical drives offline.
  - *Root cause*: When a logical drive has multiple predictive failed drives, the firmware disables the predictive failed drive avoidance policy. This can cause I/O latency issues. After all but one of the predictive drives are

replaced and the logical drive rebuilds are completed, the I/O latency issue may remain and result in the OS marking the logical drive as offline.
  - *Fix*: Ensure updating the predictive failed drive avoidance policy status after every logical volume state update.
  - *Risk*: Low
- Fixed an issue where a transforming logical drive fails when the controller is abruptly rebooted after logical drive deletion.
  - *Root cause*: If an array has multiple logical drives and if the host deletes any logical drive that is not the last in the array, then transformation is queued for all the logical drives physically located after the deleted logical drive. Each transformation request suspends the current transformation, updates the RAID metadata, saves it in logical drives, and update the transformation progress status. So when one such transformation request is received from the host, the firmware suspended the transformation and then updates saved the metadata into logical drives, and before updating the transformation progress status, the system is abruptly rebooted. This causes the logical drives' metadata and transformation progress status to be out of sync. Due to this reason, the logical drive moves to a FAILED state.
  - *Fix*: Add an extra variable in the RAID metadata to store the current transformation progress information and use it to invalidate the transformation progress data due to an abrupt reboot.
  - *Risk*: Medium
- Fixed an issue where the RAID metadata is present after clear configuration and a reboot.
  - *Root cause*: When predictive spare rebuild is completed on a logical drive, firmware fails the predictive failed drive. But firmware did not clear the RAID metadata present in the failed drive during clear configuration. During the next boot, the firmware found valid RAID metadata in the predictive failed drive and loads the RAID metadata causing the logical drive to come up again.
  - *Fix*: Firmware will spin up the failed drive during clear configuration and clear the RAID metadata.
  - *Risk*: Low
- Fixed an issue where the controller locked up while resuming transformation after a cold boot.
  - *Root cause*: If a transformation running with the internal-cache method was interrupted due to abrupt power loss, and the backup power source is removed on the next boot, the transformation method switches from the internal-cache to a disk-based method. The firmware copies the data from the internal cache to disk and during this process an uninitialized variable was used that triggered a firmware lockup.
  - *Fix*: Firmware will initialize the variable before utilizing it.
  - *Risk*: Low
- Fixed an issue an invalid persistent event tag ID is returned for reboot marker event.
  - *Root cause*: The firmware fails to calculate the proper persistent event tag value for the reboot marker event.
  - *Fix*: Store the persistent event tag value in the persistent firmware metadata. On boot, the persistent event tag value is read from the persistent firmware metadata to resume the numbering from the previous boot.
  - *Risk*: Low
- Fixed an issue where a persistent event gave invalid data in the timestamp field.
  - *Root cause*: During boot, firmware has not yet received the current time from the Real-Time Clock, which causes the persistent event log boot marker event to have invalid data in the timestamp field.
  - *Fix*: Reset the event structure to zero so it provides zeroes instead of invalid data in the timestamp field.
  - *Risk*: Low
- Fixed an issue where the logical drive failed during the heal array transformation.
  - *Root cause*: When the firmware receives heal array operation on a logical drive with an active spare, the transformation begins. During this transformation, if a previously failed disk is replaced, firmware incorrectly releases the active spare before completing the transformation and moves the logical drive to FAILED state.
  - *Fix*: Avoid releasing an active spare disk during the heal array transformation. After the transformation is completed, it gets released to the respective spare pool.
  - *Risk*: Medium
- Fixed an issue where rebuild does not start on an array with the spare disk.

  - *Root cause*: When firmware receives heal array operation on a logical drive with an active spare, the transformation begins. Once the transformation is completed, firmware failed to release the active spare from the old configuration. Due to this, the spare disk is in a Used state, and firmware is unable to utilize this spare disk for rebuild operation on the new configuration.
  - *Fix*: When a transformation is completed, firmware must release the spare disks associated with the old configuration.
  - *Risk*: Medium
- Fixed an issue where the debug log was unable to capture persistent event logs.
  - *Root cause*: The firmware moved the persistent event log pointer ahead of the logged events. Due to this, firmware fails to capture the logged events in the persistent event log memory.
  - *Fix*: Ensure the firmware never moves persistent event log pointer ahead of the logged events.
  - *Risk*: Low
- Fixed a possible lockup that could happen during bootup on a logical drive that was undergoing transformation and reboot happens during transformation, and then a drive fails while battery/super capacitor finishes charging.
  - *Root cause*: Logical drive gets into Abnormal state. As part of this process, a call was made to restart transformation, but another thread process had marked transformation already running, so this resulted in lockup.
  - *Fix*: Made fix to suspend transformation when putting logical drive into Abnormal state.
  - *Risk*: Low
- Fixed an issue where volatile key support was showing enabled when remote encryption was being used.
  - *Root cause*: When enabling remote encryption, a bit was not being cleared that advertised volatile key support enabled.
  - *Fix*: Clear this volatile key support bit when enabling remote encryption.
  - *Risk*: Low
- Fixed an issue where PLDM event consumer was missing events from previous boot.
  - *Root cause*: PLDM event consumer was initialized before events were put into buffer.
  - *Fix*: Ensure PLDM event consumer is initialized after events are placed in buffer.
  - *Risk*: Low
- Fixed an issue where the controller would lock up on the first boot if local NVRAM is populated with all FFs.
  - *Root cause*: The checksum did not fail and the value of FF for version was larger than expected.
  - *Fix*: If local version is 0xFFFF, then set local NVRAM to defaults.
  - *Risk*: Low
- Fixed an issue where the backup power source was missing reporting the maximum temperature and temperature threshold.
  - *Root cause*: Code was missing to fetch these values.
  - *Fix*: Added code to report these values.
  - *Risk*: Low
- Fixed an issue where a data loss could be seen when migrating drives involving IOBypass from one controller to another with a different DDR cache module size.
  - *Root cause*: Pointers were not set up properly to handle the change in cache size of new controller.
  - *Fix*: Fixed pointers to be updated properly based on size of the cache present.
  - *Risk*: Low
- Fixed an issue where the SCSI-to-NVME translation layer incorrectly filtered some VERIFY commands as unsupported commands leading to the firmware locking-up.
  - *Root cause*: When the firmware translates SCSI commands, it ensures the commands are supported. If the firmware does not support the command, it asserts. In this case, the firmware will assert on the VERIFY_10, VERIFY_12, and VERIFY_16 commands even though the firmware should support these command.
  - *Fix*: Add the VERIFY_10, VERIFY_12, and VERIFY_16 command to the list of commands the FW will support for SCSI-to-NVME translation.
  - *Risk*: Low
- Fixed a lock-up due to the firmware accessing an unbound logical port during link state change.

  - *Root cause*: In some scenarios, a logical port may become unbound during a link state change before the firmware accesses it to retrieve information needed to report the link state change. By unbinding a port, the firmware will free all the resources associated with the logical port. If the firmware attempts to access an unbound port, it will access unallocated data which may result accessing invalid addresses.
  - *Fix*: Add a check to confirm that a port is bound to a PHY. The firmware only accesses the port information if the port is bound.
  - *Risk*: Low
- Added firmware workaround for Intel PCIe link down testing of upstream port (USP) causing PCIe LTSSM failures.
  - *Root cause*: The firmware thread responsible for re-enabling the LTSSM on a link up event is at a lower priority relative to other threads handling the link up processing. This results in a delay in re-enabling the LTSSM. The extra delay violates the PCIe spec timing requirement. If in the Polling.Configuration state for greater than 24ms, timeout and move to Polling.Compliance. PCIe traces showed the controller timing out (>24 mS) with no transition to Polling.Compliance.
  - *Fix*: Dynamically increase the priority of the thread responsible for re-enabling the LTSSM on a link-down event and return the thread's priority to its original level after the LTSSM has been re-enabled when processing the link-up event.
  - *Risk*: Medium
- Return valid sense data for NVME drives that do not support Sanitize.
  - *Root cause*: NVMe drives that did not support Sanitize would not return valid sense data on a Request Sense SCSI command.
  - *Fix*: If a drive does not support Sanitize, return valid sense data and ensure that drives that were previously failed and recovered can support it.
  - *Risk*: Low
- Fixed an issue where invalid characters were written to the persistent event log.
  - *Root cause*: The persistent event log is stored in a Ferroelectric RAM (FRAM), which is a non-volatile device. Due to differences in the write behavior on some FRAM parts, the persistent event log data will not be written to the device correctly if the write operation crosses the device’s page boundary.
  - *Fix*: Firmware will ensure persistent event log data that crosses the device’s page boundary is written correctly.
  - *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 2.4.1/Legacy BIOS Build 2.4.3

This release includes the following UEFI fixes and enhancements:

- Added an HII option to configure Persistent Event Log Policy.
- Added an HII field to show Consistency Check status for each logical drive.
- Fixed an issue where system freezes when 64 failed logical drives are present.
  - *Root cause*: While framing driver health message with error code 0x1946, the length of message copied is more than the allocated amount causing the system to access invalid memory.
  - *Fix*: Allocate and copy the message as per the content of dynamic message.
  - *Risk*: Low
- Fixed an issue driver health message with error code 0x1943 is not shown when expected.
  - *Root cause*: Controller reporting unsupported status is not considered when encrypted devices are connected to an unsupported controller.
  - *Fix*: Trigger 0x1943 error message when controller reports unsupported configuration.
  - *Risk*: Low
- Fixed an issue where OS on a logical drive does not boot when an intermediate logical drive within the array is deleted.
  - *Root cause*: Block I/O calls fail due to incorrect internal indexing to route the I/O when logical drive numbering is not linear.

  - *Fix*: Corrected logical drive indexing to consider the actual logical drive number.
  - *Risk*: Low
- Fixed an issue where no logical drive information is shown when Managed SED controller password is set.
  - *Root cause*: storageCore blocks any editable configuration APIs when controller password is enabled.
  - *Fix*: Change to direct APIs instead of editable configuration APIs for populating logical drive information.
  - *Risk*: Low
- Fixed an issue where the controller firmware updated with a wrong image returns success even if it is failing.
  - *Root cause*: Error returned from internal command buffer is not mapped with the controller error status.
  - *Fix*: Propagate error infomation from the low-level command interface to the top.
  - *Risk*: Low
- Fixed an issue where Block I/O calls to multi-LUN devices fails if the multi-LUN configuration is changed.
  - *Root cause*: Multi-LUN re-enumeration in HII caused clearing of the index data.
  - *Fix*: Do not re-enumerate multi-LUN devices in HII as it is not required.
  - *Risk*: Low

2.1.3 Driver Fixes

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

2.1.3.1 Windows Driver Fixes

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

2.1.3.1.1 Fixes and Enhancements for Windows Driver Build 1010.52.0.1012

This release provides the following fixes and enhancements.

- Fixed an issue where I/O errors are observed in a multipath configuration when cable is unplugged/plugged.
  - *Root cause*: The SmartPQI driver returns I/O with the following SRB status and SCSI check conditions, which leads the Disk/MPIO driver to report disk errors on multipath configuration.

```
SrbStatus = SRB_STATUS_ERROR
    ScsiStatus= Check Condition
    SenseKey  = 0x05 Illegal Request
    ASC&ASCQ  = 26:00 Invalid Field In Parameter List or
                25:00 Logical Unit Not Supported
```

  - *Fix*: The SmartPQI driver processes the I/O with SCSI error `Sense Key: Illegal Request` on multipath physical devices and then needs to return `SrbStatus=SRB_STATUS_NO_DEVICE` instead of `SrbStatus=SRB_STATUS_ERROR` allowing the MPIO driver to perform I/O failover.
  - *Risk*: Medium
- Fixed an issue where the default driver setting does not properly set the drive queue depth. This is seen with physical drives and logical drives after a hot-plug event.
  - *Root cause*: The SmartPQI driver added multi-LUN drive support and code was added that set all the LUNs off a specified target. Driver was passing wrong BTL address to Storport API set queue depth.
  - *Fix*: Pass the correct (bus, TargetId, Lun) address for the device.
  - *Risk*: Low
- Fixed an issue where the SmartPQI driver is not loading with VMs.
  - *Root cause*: On a VM Server 2016 with Discrete Device Assignment (DDA) due to a February security update the PCI command register bit PCI_ENABLE_MEMORY_SPACE (0x0002) does not get explicitly set when the underlying bus driver is VPCI(that is, in a VM). This caused the SmartPQI driver not to load due to the driver checking for the bit to be set.
  - *Fix*: Removed checking the PCI command register bit PCI_ENABLE_MEMORY_SPACE (0x0002) so the driver will load. The underlying bus driver already guarantees that access to the device MMIO registers is enabled.
  - *Risk*: Low
- Fixed an issue where the Diskpart utility shows one disk's SAN policy is offline after updating the device driver.
  - *Root Cause*: The device driver assigns a new SCSI Target ID to the last disk of the SES/SEP group that causes PartMgr to detect it as a new device and set it offline.

- *Fix*: The driver assigns the same SCSI Target ID for all devices within the SES/SEP group.
- *Risk*: Low

2.1.3.2 Linux Driver Fixes

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

2.1.3.2.1 Fixes and Enhancements for Linux Driver Build 2.1.20-035

This release provides the following fixes and enhancements.

- Switched to using "blk-mq" tags instead of linear searching.
- Fixed an issue where the maximum LUN number supported by SmartPQI is now set correctly.
  - *Root cause*: When multi-actuator support was added to SmartPQI, the maximum number of LUNs supported by SmartPQI was supposed to be changed from unlimited to 256, but the setting was inadvertently left at unlimited.
  - *Fix*: The maximum LUN number supported by SmartPQI is now set correctly to 256.
  - *Risk*: Low
- Fixed an issue where Linux performance drops when large CPU affinity is used.
  - *Root Cause*: The driver was using a single hint variable in the function that gets a free I/O request element from the I/O request pool that was causing contention when it was utilized by a large number of threads.
  - *Fix*: Eliminate the initial contention by removing the hint and instead assign each CPU its own starting point within the request element array based on its CPU number.
  - *Risk*: Low
- Fixed an issue to update hardware queue mapping when “block-mq” is not enabled or supported.
  - *Root cause*: No mapping for CPUs exceeding the maximum queue group count.
  - *Fix*: Updated the mapping algorithm to provide a valid mapping for all CPUs.
  - *Risk*: Low
- Fixed an issue where “block-mq” and managed interrupts support are not enabled by default for 5.x Linux kernels.
  - *Root cause*: The appropriate definitions are not enabled in the build files.
  - *Fix*: Enable the appropriate flags for 5.x Linux kernels.
  - *Risk*: Low
- Fixed a problem where the driver does not issue flush cache to physical drives during PCIe hot remove.
  - *Root cause*: During controller PCIe graceful hot remove, the driver does not send commands to the drives to flush the cache.
  - *Fix*: Add Graceful Removal state check in remove path to allow flush cache to be issued to the physical drives.
  - *Risk*: Low
- In some situations, the presence of a multi-actuator drive could cause no drives to be listed for a controller, during OS installation. The driver can also hit an unrecoverable Call trace during rmmod.
  - *Root cause*: The `pqi_slave_destroy` routine is called multiple times for a multi-LUN device that causes a Call trace.
  - *Fix*: Remove device only upon the last `pqi_slave_destroy` call.
  - *Risk*: Low

2.1.3.3 VMware Driver Fixes

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

2.1.3.3.1 Fixes and Enhancements for VMware Driver Build 4380.0.108

This release provides the following enhancements and fixes:

- Fixed an issue of PSOD when deleting logical drive.
  - *Root Cause*: Driver maintains a linked list of removed devices protected by a lock. A timer function iterates through the list and frees those devices whose timeout has expired. In the timer function, the lock has been released for a short interval to notify the OS about device removal and acquired back. During this time, the device that was in removal stage came back. Due to this, driver removed this entry from the list. The timer function was trying to free this entry from list which was already freed.

  - *Fix*: Check for the device removal state before freeing the device entry.
  - *Risk*: Low
- Fixed an issue where firmware lockup was observed.
  - *Root cause*: For a RAID logical drive with IOBypass enabled, driver frames the CDB based on the SCSI command. This may result in translating a SCSI READ(16) to a READ(10) based on the LBA and transfer size and set the command length to 10 bytes. If such I/O fails, the driver will retry the same command via RAID path. The driver was supposed to use the original CDB sent by the OS SCSI layer, but the driver does not reset the command length back to the original command length of 16 bytes. This may result in a partial copy of the SCSI CDB with incorrect CDB transfer length. Firmware will frame the SGL based on CDB transfer length which results in SGLs that do not match with the data transfer size, and that triggers a firmware lockup.
  - *Fix*: Reset the SCSI command length when retrying an IOBypass IO down the RAID path.
  - *Risk*: Medium

2.1.3.4 FreeBSD/Solaris Driver Fixes

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

2.1.3.4.1 Fixes and Enhancements for FreeBSD Driver Build 4330.0.1038

There are no known fixes for this release.

2.1.3.4.2 Fixes and Enhancements for Solaris Driver Build 11.4120.0.1005

There are no fixes and enhancements for this version.

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 4.09.00.25611

This release provides the following fixes and enhancements.

- Added ESXi 8.0 support for maxView and ARCCONF.
- Added a display property in arcconf GETCONFIG and GETVERSION command ouput to display the SEEPROM version.
- Added support in ARCCONF to update the PSOC expander firmware.
- Added an option in maxView and ARCCONF to configure the Persistent Event Log Policy.
- Added support in maxView and ARCCONF to configure remote Key Management Server (KMS) based CBE (Controller-Based Encryption).
- Added properties in maxView and ARCCONF to display the logical device consistency check runtime metrics.
- Added following UI enhancements in maxView:
  - Added an option to switch the ribbon between classic and simplified view. The simplified view displays only the applicable operations in the ribbon.
  - Added a new **Inventory** tab in maxView enterprise node to display and export the configurations in a CSV format.
  - Added a new **Properties** tab in maxView physical device node and moved few properties from **Summary** tab for better user experience.
  - Consolidated all the resources related properties from other tabs to **Resources** tab in physical device and logical device node.
- Fixed an issue where UEFI ARCCONF was not allowing the user to enable the erase completed drive.
  - *Root cause*: Operation to enable erased drive was not available in UEFI ARCCONF due to a wrong feature bit check.
  - *Fix*: Feature bit check is corrected to allow enabling erased drive operation in UEFI ARCCONF.
  - *Risk*: Low
- Fixed an issue where UEFI ARCCONF displayed unreadable text for reported location in GETCONFIG command output.

  - *Root cause*: Invalid format specifier was used for reported location string in display.
  - *Fix*: Added changes to use the valid format specifier to display the reported location in UEFI ARCCONF GETCONFIG command output.
  - *Risk*: Low
- Fixed an issue where maxView was not allowing to select the 'Number of Targets' drop-down for backplane discovery protocol.
  - *Root cause*: The empty list was returned for "Number of Targets" that resulted in non-selectable drop-down during backplane discovery protocol change in maxView.
  - *Fix*: The overwritten empty list is removed and returned with valid values for "Number of Targets". Now, the "Number of Targets" drop-down is selectable and valid values are listed in maxView.
  - *Risk*: Low
- Fixed an issue in maxView where **Management Protocol** drop-down was not relevant after depreciation of CIM protocol in ESXi 7.x and above.
  - *Root cause*: **Management Protocol** drop-down was added in maxView when both CIM and redfish were supported. CIM is no longer supported by maxView. So, the **Management Protocol** is not applicable anymore.
  - *Fix*: Removed **Management Protocol** and added **Operating System** drop-down in **Add System** dialog with options **Windows/Linux**, **ESXi 7.x**, and **ESXi 8.x**.
  - *Risk*: Low
- Fixed an issue in maxView where the connector level mode change was allowed when the controller was waiting for the adapter password whereas the same operation was blocked at the controller level.
  - *Root cause*: The Connector Level mode change was allowed from maxView when the controller was waiting for the adapter password. When the controller was waiting for password the Connector mode change should be blocked from controller and individual connector level.
  - *Fix*: The Connector mode change is blocked from maxView when the controller is waiting for adapter password. maxView blocks this operation from both Controller and Connector level.
  - *Risk*: Low
- Fixed an issue in maxView where the Revert to OFS operation was not working when the PSID of SED drive was entered in lowercase.
  - *Root cause*: maxView was not allowing next step when PSID was in lowercase during Revert to OFS operation. There was a check in maxView to allow only uppercase PSID.
  - *Fix*: The check for validating PSID in maxView is modified to accept both uppercase and lowercase PSID for Revert to OFS operation.
  - *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 6.15.13.0

This release provides the following fixes and enhancements.

- Redfish GET on the VolumeCapabilities resource will now include the annotation Name@Redfish.OptionalOnCreate. Additionally, the Redfish GET response for a Volume resource shows both the **DisplayName** and **Name** properties published with the same value, that is, the volume label. Either of **DisplayName** or **Name** can be used to set a volume label at create time using Redfish POST or modify the label using Redfish PATCH.
- Redfish GET on a drive resource in PLDM will now support Multi-Actuator (MA) drives. There will be a single drive resource per MA drive with CapacityInBytes being equal to the total capacities of all the LUNs. The Identifiers include the **DurableName** and **DurableNameFormat** of each LUN. All other properties are the same.
- In addition to the existing NumericSensor PDR published for the hard drive temperature sensor, a new set of NumericSensor PDRs with entityInstanceNumber = 2 has been added to provide temperature readings for individual drives.
- The Type 5 commands QueryDownstreamDevices, QueryDownstreamIdentifiers, and GetDownstreamFirmwareParameters now provide information for enclosure SEPs connected to the controller. The following descriptors will be reported for the enclosure SEPs using the QueryDownstreamIdentifiers command:

  - SCSI Vendor ID
  - SCSI Product ID
  - Vendor-defined descriptor containing the SEP location in “Slot=<slot>:Port=<port>:Box=<box>” format.
- Redfish POST requests to perform the Drive.Actions.#SecureErase ACTION will now be rejected with the extended error message ResourceInUse if the targeted Drive is a SED that is not in the original factory state (OFS).
- Removed the Status.Health property from Redfish GET responses for the Storage resource.
- Sending a Redfish PATCH request to update the WriteCacheEnabled property of a Drive resource is now supported for drives configured as a Volume's data drive.
- Fixed an issue where an unnecessary DriveOK alert was sent when importing a foreign SED.
  - *Root cause*: The logic for the drive alerts code was checking for DriveOfflineCleared conditions in the same If condition for the DriveOK alert.
  - *Fix*: Fixed the drive alert generation logic to remove the unnecessary DriveOfflineCleared check.
  - *Risk*: Low
- Fixed an issue where a Redfish PATCH to set Volume.IOPerfModeEnabled to true silently disabled caching for other volumes on the same array.
  - *Root cause*: The code was combining the check for other volume cache policies with another check that could cause a different error. This caused the check to let the requests through.
  - *Fix*: Made the two checks independent of each other and now fail such calls with the appropriate error information.
  - *Risk*: Low
- Fixed an issue where not all permitted values were published for the VolumeCapabilities WriteCachePolicy@Redfish.AllowableValues.
  - *Root cause*: The condition for publishing allowable WriteCachePolicy values was not accounting for cases where controller caching is supported but not configured.
  - *Fix*: Added a check for the controller cache not being configured when determining the WriteCachePolicy@Redfish.AllowableValues values to publish.
  - *Risk*: Low
- Fixed an issue where a Type 5 GetStatus command following an ActivateFirmware command might return the wrong ReasonCode.
  - *Root cause*: ReasonCode field was not set correctly when the firmware device proxy processes the ActivateFirmware command resulting in an incorrect value.
  - *Fix*: Firmware device proxy now correctly sets the ReasonCode field in the GetStatus response to “1—ActivateFirmware command was received” after the update agent sends an ActivateFirmware command for a downstream device (drive).
  - *Risk*: Low
- Fixed an issue where the VolumeCapabilities resource can be published with incorrect values for WriteCachePolicy@Redfish.AllowableValues when the controller's battery is charging.
  - *Root cause*: In cases where the backup power source is charging, the cache goes into a temporarily degraded state. In such case, the VolumeCapabilities READ response uses the logic intended for unconfigured cache to determine the WriteCachePolicy allowable values.
  - *Fix*: Modified the VolumeCapabilities READ response generation code to determine the allowable values for WriteCachePolicy using configured cache rules when the cache is temporarily disabled.
  - *Risk*: Low
- Fixed an issue where the ControllerPasswordEntered Redfish alert was sent with an incorrect messageId.
  - *Root cause*: The ControllerPasswordEntered Redfish alert changed to ControllerPasswordAccepted in the released version of the DMTF StorageDevice registry v1.1.
  - *Fix*: Changed the ControllerPasswordEntered Redfish alert to ControllerPasswordAccepted.
  - *Risk*: Low
- Fixed an issue where the Links.Storage property was not published with the drive resource.
  - *Root cause*: Links.Storage was not included in the drive resource schema dictionary, and no implementation was present to publish that property with the drive resource.

  - *Fix*: Updated the schema dictionaries to include Links.Storage in the drive resource and added the property to the drive resource Redfish GET response.
  - *Risk*: Low
- Fixed an issue where the StorageController's CacheSummary.Status property is published with a value of StandbyOffline instead of UnavailableOffline when the backup power source cable is not connected to the controller at boot.
  - *Root cause*: No specific check for a missing backup power source was included when determining the StorageController's CacheSummary property values.
  - *Fix*: Modifed the StorageController RDE GET handler to publish CacheSummary.Status with a value of UnavailableOffline in the cited case.
  - *Risk*: Low
- Fixed an issue where the StorageController's Status.Health property is published with a value of Warning instead of OK when the backup power source cable is not connected to the controller at boot.
  - *Root cause*: No specific check for a missing backup power source was included when determining the StorageController's Status.Health property values.
  - *Fix*: Modifed the StorageController RDE GET handler to publish Status.Health with a value of Ok in the cited case.
  - *Risk*: Low
- Fixed an issue where a drive's ServiceLabel is sometimes erroneously published with a leading zero on its port number.
  - *Root cause*: Current logic was formatting the port in the ServiceLabel as “%02u” which results in the leading 0 when the port is a single digit number.
  - *Fix*: Corrected the logic to make the formatting of the port in the ServiceLabel dynamic based on the string length of the port name.
  - *Risk*: Low
- Fixed an issue where GetDownstreamFirmwareParameters returned incorrect ComponentActivationMethods and CapabilitiesDuringUpdate values.
  - Root Cause: A bit indicating support for drive firmware updates was not being set.
  - Fix: Firmware updates for downstream devices have been enabled for UBM and drive devices. Since UBM firmware updates are not allowed on some controllers, a check has been put into place to verify if the UBM firmware updates are allowed when attempting to update the device.
  - Risk: Low
- Fixed an issue where Redfish alerts generated early in the boot sequence were not being received by the Server Management controller.
  - *Root cause*: Publishing Redfish events required event support to be negotiated using NegotiateRedfishParameters. This negotiation was being done after the initial polling for controller events, so those initial Redfish events were not sent to the management controller.
  - *Fix*: Removed the requirement for event support negotiation as a prerequisite for passing Redfish events.
  - *Risk*: Medium
- Fixed an issue where a volume create request is erroneously accepted when the request payload contains a WriteCachePolicy value conflicting with the existing cache configuration.
  - *Root cause*: Volume create request payload validation was missing a check for requests for active write caching when the controller's cache ratio setting was set to 100% reads.
  - *Fix*: Added a validation check to reject such requests with a messageId of PropertyValueIncorrect.
  - *Risk*: Low
- Fixed an issue where the ControllerPreviousError alert was sent with an incorrect messageId and severity.
  - *Root cause*: The internal table of alert definitions includes MessageId and Severity information for this alert that is not compliant with the DMTF alert registry.
  - *Fix*: Updated the alert definitions to include correct MessageId and Severity information.
  - *Risk*: Low
- Fixed an issue where the BatteryCharging alert was sent with an incorrect severity on certain controllers.
  - *Root cause*: The BatteryCharging alert was hard-coded to only be sent with a Severity of Warning.
  - *Fix*: Modified the BatteryCharging event logic to send a Severity of OK for certain controllers.

- *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)
  - No Storage Manager support

2.2.2 Firmware Limitations

This section shows the firmware limitations for this release.

2.2.2.1 Limitations for Firmware Release 03.01.17.56

This release includes the following limitations.

- Persistent Event Logs (PEL) will be cleared under the following conditions:
  - Upgrading to the firmware version 03.01.17.56.
  - Downgrading from the firmware version 03.01.17.56.
- Flashing a new firmware release to a controller running in VMware's Passthrough mode and triggering a VM reset will not cause the new firmware to run. The controller will still be running the previous firmware.
  - *Workaround*: Restart the server itself to cause the controller to load the new firmware.

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 2.4.1/Legacy BIOS Build 2.4.3

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 Windows Driver Limitations

This section shows the Windows driver limitations for this release.

2.2.4.1.1 Limitations for Windows Driver Build 1010.52.0.1012

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, example of 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. 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.
- In certain circumstances the installation may fail on Windows Server 2016 and Windows 2012 R2 after selecting drives.
  - *Workaround*: Follow these steps to ensure drives are clean and all partitions are removed before beginning a new installation.
    - Hit Shift + F10 to Open Windows prompt

  - Type **diskpart**
  - List disk
  - Select the disk you want to clean
  - Clean
- BSOD observed on Windows Server 2019 while loading OOB driver if BIOS setting SNC4 is enabled.
  - *Workaround*: Change BIOS setting to SNC2.

2.2.4.2 Linux Driver Limitations

This section shows the Linux driver limitations for this release.

2.2.4.2.1 Limitations for Linux Driver Build 2.1.20-035

This release includes the following limitations.

- This release includes the following limitation when doing a driver injection (DUD) install. On some distributions (RHEL7.9, RHEL8.2, RHEL8.3, SLES15SP2, SLES15SP3, OpenEuler 22.03LTS), the DUD install will hang if an attached drive (either HBA mode or logical drive) has Write Cache enabled.
  - *Workaround*: There are two workarounds for this issue:
    - Make sure the Write Cache is disabled for any attached drive.
    - For RHEL7.9/8.2/8.3 and OpenEuler 22.03LTS, add `rd.driver.blacklist=smartpqi` to the grub entry along with inst.dd.
- RHEL 8.7 - Unable to inject OOB driver (DUD) on RHEL 8.7 when NVMe drives attached to the system. This is a multipath issue with the OS install process.
  - *Workaround*: Edit grub to include the boot argument `nompath`. Therefore, 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. Reported on RHEL 8.5, 8.6,9.0 and 9.1.
  - *Workaround*: There are two workarounds for this issue:
    - Load OS from USB device instead of virtual media.
    - Load OS from virtual media but initiation ISO verification (media test) during install followed by ESC to cancel media test.
- 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 is expected in the mid-October UEK7 release, version number still pending.
  Note: This does not affect Oracle 8 UEK 7.
  - *Workaround*: Install the rpm using `--nodeps` when dependency failures occur.
    - 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, install the SmartPQI rpm using the `--nodeps`.
- On AMD/RHEL 7.9 systems, the system might panic due to a bug in the IOMMU module. For details, see https://lore.kernel.org/linux-iommu/20191018093830.GA26328@suse.de/t/
  - *Workaround*: Disable the IOMMU setting option in BIOS.
- Depending on hardware configurations, the SmartPQI `expose_ld_first` parameter may not always work consistently.
  - *Workaround*: None
- 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:
    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 XenServer 8.1 LTS, Ubuntu 18.04.5 LTS), only one multi-actuator drive LUN is displayed in the OS installation menu.
  - *Workaround*: Inject/Load the OOB driver during OS installation. Go to console mode (Ctrl+Alt+F2), issue the command “rmmod smartpqi” followed by “modprobe smartpqi”. Exit console mode (Ctrl+Alt+F1) and proceed to the Primary disk selection screen in the GUI.
- On some distributions (including RHEL 9.0/Oracle Linux 9.0), you are unable to inject the OOB driver (DUD) when a multi-actuator drive is attached.
  – *Workaround*: Install using the inbox driver, complete OS installation, then install OOB driver.

2.2.4.3 VMware Driver Limitations

This section shows VMware driver limitations for this release.

2.2.4.3.1 Limitations for VMware Driver Build 4380.0.108

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, you can also 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 an single processor AMD system. Lockup has only been seen with in a Linux Guest VM. No known workaround at the present time.

2.2.4.4 FreeBSD/Solaris Driver Limitations

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

2.2.4.4.1 Limitations for FreeBSD Driver Build 4330.0.1038

This release includes the following limitation.

- FreeBSD OS installation on a NVMe physical device is not supported in this release. (FreeBSD 13.0)

2.2.4.4.2 Limitations for Solaris Driver Build 11.4120.0.1005

There are no known limitations for this release.

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 4.09.00.25611

There are no known limitations for this release.

2.2.5.2 PLDM Limiations

This section shows the PLDM limitations for this release.

2.2.5.2.1 Limitations for PLDM Release 6.15.13.0

This release includes the following limitation.

- Action Storage.ResetToDefault with a ResetType of 'ResetAll' is not supported when the controller has volumes which are encrypted.

3. Updating the Controller Firmware

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

3.1 Updating Controllers to Latest Firmware

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

3.1.1 Upgrading to 3.0X.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
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.

The Microchip Website

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

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

Product Change Notification Service

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

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

Customer Support

Users of Microchip products can receive assistance through several channels:

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

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

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

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip 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.

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.

Trademarks

ISBN: 978-1-5224-9555-0

Quality Management System

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

Worldwide Sales and Service

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