

Release Notes

HBA 1100 Software/Firmware

Released
October 2019



a  MICROCHIP company

Revision History

Revision	Revision Date	Details of Change
15	October 2019	Update for SR 2.5
14	August 2019	Update for SR 2.4.8 Release
13	March 2019	Update for SR 2.4.4 Release
12	January 2019	SR2.4 Production Release
11	October 2018	SR2.3 firmware update with Cavium/ARM support and Ubuntu driver.
10	June 2018	SR2.3 Production Release
9	October 2017	Update Supported OSs
8	October 2017	First Production Release
1-7	December 2017 - July 2017	Pre-Production Releases

Contents

1 About This Release.....	4
1.1 Release Identification.....	4
1.2 Components and Documents Included in this Release.....	5
1.3 Files Included in this Release.....	6
2 What is New?.....	9
2.1 Features.....	9
2.2 Fixes.....	9
2.2.1 Firmware Fixes.....	9
2.2.2 UEFI Fixes.....	11
2.2.3 Driver Fixes.....	12
2.2.4 Management Software Fixes.....	14
2.3 Limitations.....	14
2.3.1 Firmware Limitations.....	14
2.3.2 UEFI Limitations.....	15
2.3.3 Driver Limitations.....	15
2.3.4 Hardware Limitations.....	16
2.3.5 Management Software Limitations.....	16
3 Updating the Board Firmware for PQI Operation.....	17
3.1 Updating Controllers to latest (PQI) Firmware.....	17
4 Installing the Drivers.....	18

1 About This Release

The development release described in this document includes firmware, OS drivers, tools, and host management software for the HBA 1100 Series adapter solutions from Microsemi.

1.1 Release Identification

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

Table 1 • Release Summary

Solutions Release	2.5
Package Release Date	October 15, 2019
Firmware Version	2.62 B0 ^{1,2} (basecode 06.05.002.001)
UEFI Version	1.3.9.2
Legacy BIOS	1.3.9.2
Driver Versions	Windows SmartPQI: <ul style="list-style-type: none"> • Windows 2012/2016/2019: 106.166.0.1022 • Windows 7/2008: 6.100.0.1014 Linux SmartPQI: <ul style="list-style-type: none"> • RHEL 6/RHEL 7/RHEL 8/SLES 12/SLES 15: 1.2.10-025 • Ubuntu 16/18: 1.2.10-025 • CentOS 6/7/8: 1.2.10-025 • Debian 8/9: 1.2.10-025 VMware SmartPQI: <ul style="list-style-type: none"> • VMWare ESXi 6.0/6.5/6.7: 1.0.4.3008 FreeBSD/Solaris SmartPQI: <ul style="list-style-type: none"> • FreeBSD 11/12: 1.0.4.3008 • Solaris 11: 1.0.4.3008
arcconf/Maxview	B23668

Note:

1. Downgrading to 1.04 B0 or older builds from this release or prior 1.29 releases may cause the board to not boot or have supercap errors due to an incompatibility in SEEPROMs between this release and prior releases. Refer to the section " [Updating the Controller Firmware](#) " to downgrade an existing board.
2. If the firmware running on the board is older than 0.01 B594, existing data in the logical volumes must be backed up if it needs to be used after the upgrade. After the upgrade from firmware prior to 0.01 B594, the logical volumes will need to be recreated.
3. Only run the driver on firmware 0.01 build 500 or later.

1.2 Components and Documents Included in this Release

Download the firmware, drivers, host management software, and supporting documentation for your HBA1100 controller solution from the Microsemi Web site at <https://storage.microsemi.com/en-us/support/start/>

1.3 Files Included in this Release

This release consists of the files listed in the following tables:

Firmware Files

Table 2 • Firmware Files

Component	Description	Pre-Assembly Use	Post-Assembly Use
SmartFWx100.bin	Programmable NOR Flash File Use to program NOR Flash for boards that are already running firmware.		X

Table 3 • Firmware Programming Tools

Tool	Description	Executable
Arccconf romupdate	The command allows to upgrade/downgrade the firmware and BIOS image to the controller.	Refer to Table 7 • Host Management Utilities on page 8
maxView firmware upgrade wizar- ard	The firmware upgrade wizard allows to upgrade/downgrade the firmware and BIOS image to one or more controller(s) of same model in the system.	Refer to Table 7 • Host Management Utilities on page 8

Driver Files

Table 4 • Windows Storport Miniport SmartPQI Drivers

Package	Drivers	Binary	Version
2012	Server 2019 Server 2016 and Windows 10 Server 2012, R2 and Windows 8.1, 8	SmartPqi.sys	x64
		SmartPqi.inf	x64
		Smartpqi.cat	x64
2008	Server 2008 R2 SP1 and Windows 7	SmartPqi.sys	x64
		SmartPqi.inf	x64
		SmartPqi.cat	x64

Table 5 • Linux SmartPQI Drivers

Drivers	Intel/AMD x64	Cavium ThunderX2 ARM x64 ³
Red Hat Enterprise Linux/CentOS 8.0	X	
Red Hat Enterprise Linux/CentOS 7.6, 7.5 ¹ , 7.4	X	X
Red Hat Enterprise Linux/CentOS 7.7, 7.3	X	
Red Hat Enterprise Linux/CentOS 6.10, 6.9 ¹	X	

Drivers	Intel/AMD x64	Cavium ThunderX2 ARM x64 ³
SuSE Linux Enterprise Server 12 ¹ , SP4, SP3 ²	X	X
SuSE Linux Enterprise Server 12 ¹ , SP2	X	
SuSE Linux Enterprise Server 15, SP1 ¹	X	X
Oracle Linux 7.7 UEK5u2, 7.6 UEK5u2, 7.5 UEK4	X	
Oracle Linux 7.5 with UEK4 (4.1.12-124)	X	
Oracle Linux 7.6 with UEK5u2 (4.14.35)	X	
Oracle Linux 7.7 with UEK5u2 (4.14.35)	X	
Oracle Linux 8.0	X	
Ubuntu 18.04.3 (LTS), 18.04.2, 18.04.1	X	X
Ubuntu 16.04.5, 16.04.4	X	X
Debian 10	X	
Debian 9.9	X	
Debian 8.11	X	
Citrix xenServer 8.0,7.6	X	

Note:

1. To mitigate against the Spectre Variant 2 vulnerability, the RHEL 6.9, RHEL 7.4, RHEL 7.5 and SLES12 SP3 and higher drivers have been compiled to avoid the usage of indirect jumps. This method is known as "Retpoline".
2. SLES 12 SP3 cannot be installed on drives attached to the HBA 1100 controller in Cavium Thunder X2 systems due to a CPU and OS installation issue. This driver will support the HBA 1100 controller in Cavium Thunder X2 systems for non-boot drives. For Cavium Thunder X2 servers, if you choose to install SLES12 SP3, you have to install it on the system board's SATA controller.
3. CentOS 7.4 requires the kernel to be updated to 4.11.0-44 or later.

Table 6 • FreeBSD, Solaris, and VMware SmartPQI Drivers

Drivers	Version
FreeBSD 12.0, 11.3	x64
Solaris 11.3, 11.4	x64
VMware 6.0, 6.5, 6.7	x64

Host Management Software

Table 7 • Host Management Utilities

Description	OS	Executable
ARCCONF Command Line Utility	Windows x64 Linux x64 VMware EXSi 5.5/6.0 XenServer FreeBSD x64 Solaris x86 Linux ARM	See the Arccconf download package for the OS-applicable installation executable.
ARCCONF for UEFI		Included as part of the firmware downloadable image.
maxView Storage Manager	Windows x64 Linux x64 VMware EXSi 5.5/6.0 XenServer	See the maxView Storage Manager download package for the OS-applicable installation executable.
maxView vSphere Plugin	vCenter 5.5 and 6.0	See the VMware maxView Storage Manager download package for the OS-applicable installation executable.
Boot USB (offline or pre-boot) for ARCCONF and maxView Storage Manager	Linux x64	See the maxView BootUSB download package for the .iso file.

2 What is New?

2.1 Features

The following table lists features supported for this release.

Table 8 • Feature Summary

Feature		Supported in this Release	Future Release
UEFI Driver, Boot Support		X	
Legacy Boot Support		X	
Dynamic Power Management		X	
SMR Drive Support	Enumeration, Unrestricted Command Flow-Through	X	
	SATL Translation for HA/HM SMR Management	X	
	Identify All Drive Types	X	
Driver Support	Linux	X	
Flash Support		X	
MCTP BMC Management		X	

2.2 Fixes

2.2.1 Firmware Fixes

2.2.1.1 Fixes and Enhancements for Firmware Release 2.62 B0

This release includes the following fixes and enhancements:

- Fixed a controller 0xFFFF001 lockup problem while removing all available paths to an enclosure.
 - Root Cause: While disabling both the available paths of an enclosure by hot removing the cables, the internal book keeping structure, which tracks the active I/O module of the enclosure is not updated and index calculation for the expander information data structure becomes invalid and causes a NULL pointer access exception, while background activity is updating LED status of SES target.
 - Fix: Avoid accessing of invalid expander information structure by validating the change in active I/O module when path changes from dual path to single path or single path to none.
 - Risk: Medium
- Fixed a controller hang problem while upgrading/downgrading expander firmware continuously.
 - Root Cause: The allocated memory structures were not freed during the clean-up process of expander firmware update which results in the controller running out of memory and any thread trying to allocate memory will stall resulting in a controller hang.
 - Fix: Free-up the allocated resources during the cleanup process of expander firmware update.

- Risk: Low
- Fixed a controller 0x1E30 lockup during server reboot/shutdown.
 - Root Cause: When the SmartPQI host transport interface is reset during a server reboot, if there are still outstanding I/Os, firmware will lockup because it has determined there are still outstanding host I/Os when the reset was triggered. Firmware incorrectly included commands from out of band transport as well in this outstanding I/Os count that results in the lockup even though there are no outstanding host I/Os.
 - Fix: Do not include outstanding out-of-band (OOB) I/O count while determining outstanding host I/Os during PQI reset and shutdown.
 - Risk: Low
- Fixed MCTP issue where firmware returns more data than expected resulting in buffer overflow at the initiator.
 - Root Cause: When error response IU is sent, data payload was filled with the response IU itself which causes extra data to be sent to initiator from firmware.
 - Fix: For error responses other than under-run cases, firmware should be sending only the error response IU as such. No data should be appended.
 - Risk: Medium
- Fixed a controller hang during device discovery process when a bad drive is present.
 - Root Cause: When a bad drive is hot plugged, the initial test unit ready commands is not completed, firmware sends IT Nexus Reset that is also not completed by the bad drive. But firmware proceeds to send subsequent test unit ready commands assuming that IT Nexus Reset is complete and these subsequent TUR commands were never sent since the drive is going through reset and firmware waits for these TUR commands completion forever.
 - Fix: If there is no response for IT Nexus Reset for 30 seconds, clear outstanding commands to the drive including resets, which enables the test unit ready commands to be submitted to the drive. Eventually, these TUR commands results in a timeout and the drive is failed.
 - Risk: Medium
- Fixed an issue updating global drive cache state policy for hot added SATA drives.
 - Root Cause: When a SATA drive gets hot added the validation check points to detect the drive presence returns early and the global drive cache policy doesn't get updated.
 - Fix: Logic for hot added SATA drive presence check is modified to detect them on time and global drive cache policy changes are applied immediately.
 - Risk: Low
- Fixed an issue in dual path configuration, when one of redundant path is hot removed, path information is missing in host tools.
 - RootCause: When one path is removed in a dual path configuration, fields related to that path are cleared.
 - Fix: Path information fields are retained even after one of the paths is removed.
 - Risk: Low
- Fixed an issue handling multiple outstanding SMP reset requests from the host and/or higher layer firmware.
 - Root Cause: Lower layer firmware handles multiple resets by queuing the additional resets after the first one. This creates a scenario where if the first reset is lost, subsequent requests end up queued and made to wait till the original (lost) request completes.
 - Fix:
 1. Modify lower layer firmware to reduce the SMP timeout from 50 secs to 10 secs to recover before the upper layers timeout handler kicks in.
 2. Modify lower layer firmware to return a failed response for reset, if it has been outstanding for more than 10 secs so that the upper layer can retry.

Since a failed response is returned, there will be no resets outstanding in lower layers and the subsequent retries will be processed promptly.

- Fixed an issue with SATA Secure Erase.
 - Root Cause: When a controller SAS PHY connected to a SATA drive is toggled, firmware assumed that the SATA drive was power cycled. But this is not true when the PHY being forced down then up is between the controller and expander to which the SATA drive is connected. So, when the Secure Erase operation is in progress during this PHY toggle test, firmware incorrectly cleared the Secure Erase flag assuming the SATA drive was reset and the command did not complete. In this case, we can assume that the drive will complete the Secure Erase operation and return a completion.
 - Fix: Modify firmware to refrain from clearing the Secure Erase flag if the drive did not lose power during a controller PHY toggle.
- Fixed an issue to improve the device reset response time for successful resets.
 - Root Cause: Due to variable expander behavior in handling SMP PHY Control (link/hard reset), after getting the SMP PHY Control response, lower layer firmware waits 10 seconds before deciding if a device has recovered from the reset or not. I/O will be frozen during this time. SATA device LUN reset is translated to device reset, so LUN reset response will not be returned until the 10 second timer has expired. For devices that do not come back after a reset, waiting for 10 seconds before declaring device removal is acceptable. However, for devices that successfully recovered from a reset, it is desirable to declare the reset successful sooner, which results in I/O continuation sooner, and SATA LUN Reset getting a response sooner.
 - Fix: For expander attached device reset (which includes SATA LUN Reset), utilize multiple metrics to detect the device has completed the reset and recovered, and declare the device reset complete sooner.
- Fixed a lockup issue due to firmware's attempted use of an invalid device handle.
 - Root Cause: An issue was found when a SAS port goes down in very close (ms range) proximity to a drive reset. In this case controller firmware erroneously dereferenced a device handle which was just previously invalidated, causing a lockup.
 - Fix: Modify firmware to add checks to verify if the device handle is valid before using it.

2.2.2 UEFI Fixes

Note: Microsoft signed and secure boot is supported.

2.2.2.1 Fixes and Enhancements for UEFI Build 1.3.9.2/Legacy BIOS Build 1.3.9.2

This release includes the following UEFI fixes and enhancements:

- Fixed an issue where platform boot hang was observed while booting with a multi-LUN device connected.
 - Root Cause: Invalid memory access due to mismatch in actual memory allocated and requested size for a SCSI command.
 - Fix: Corrected SCSI request size to match allocated size.
 - Exposure: All previous releases.
 - Risk: Medium
- Fixes an issue where OS boot failed when optimized boot enabled on the platform BIOS.
 - Root Cause: Driver binding supported and start methods has wrong validation for controller handle and remaining device path.
 - Fix: Driver handle and remaining device path validation corrected for driver binding supported and start methods.
 - Exposure: All previous releases.
 - Risk: Low
- Fixed an issue where arconf EFI shell command line utility prints invalid data.
 - Root Cause: Standard print format specifiers not supported in UEFI EDK2 helper methods.
 - Fix: Format specifiers changed to UEFI EDK2 library compatible format.

- Exposure: All previous releases.
- Risk: Low

2.2.3 Driver Fixes

2.2.3.1 Fixes and Enhancements for Linux Driver Build 1.2.10-025

The fixes and enhancements in this release.

- Added support for RHEL7.7 GA, Ubuntu 18.04.3 Final, SLES12SP5 RC1, RHEL&u1 Snapshot 4, and Oracle Linux 7.7 GA.
- Fixed an issue with unique ID for physical devices. The SmartPQI driver exposes a sysfs node named “unique_id” for each device that the driver exposes to the OS. This node contains a 16-byte ID that is supposed to uniquely identify each device. The ID that the driver is returning for physical devices is not necessarily always unique.
 - Root Cause: The driver gets the unique ID from VPD page 83h by reading the 16 bytes at offset 8 of the VPD data. This works for logical devices but does not always work for physical devices.
 - Fix: The driver now gets the unique IDs from the Report Physical LUN command (RPL) instead of VPD page 83h.
 - Risk: Low
- Fixed an issue where the TMF timeout is too long (60 seconds).
 - Root Cause: The TMF timeout of 30 seconds is better than 60 seconds.
 - Fix: Shortened the TMF timeout.
 - Risk: Low
- Fixed an issue where controller hang is detected due to loss of sync during a LUN reset TMF to a drive.
 - Root Cause: Since the timeout specified by the host for LUN reset TMF request is infinite, the background thread in controller firmware becomes stuck.
 - Fix: Added support for a timeout on LUN resets. When the controller firmware passes down a LUN reset, it will use the timeout value. If the timeout is hit, the controller firmware will initiate a device reset and upon completion will fail the LUN reset request to the host. The driver will then retry the LUN reset up to 3 times regardless of the failure status.
 - Risk: Medium
- Fixed an issue where controller hangs with INQUIRY command while rebooting and also becomes unresponsive.
 - Root Cause: Currently, there is no timeout mechanism for Inquiry or Pass-through commands.
 - Fix: Introduced new timeout field in RAID IU.
 - Risk: Medium
- Fixed an issue where firmware controller lockup occurs during force reboot. During the force reboot (reboot -f), there are outstanding commands while processing PQI reset, that causes firmware controller lockup.
 - Root Cause: The controller lockup issue was seen when LUN reset and shutdown happens concurrently.
 - Fix: During system shutdown, and before issuing flush cache command, block host IOs from OS and wait for all pending IOs to complete. If there are no pending IO commands, block the lun reset from OS and wait for all the sync commands to complete.
 - Risk: Medium
- Fixed an issue where the `scsi_sysfs_add_sdev` stack trace occurs while adding device.
 - Root Cause: Call trace occurred during device discovery, after multipath target was removed.
 - Fix: `sas_phy_free` should not be called for PHYs that have been set up successfully; rather, use `sas_phy_delete()`.
 - Risk: Medium

2.2.3.2 Fixes and Enhancements for FreeBSD Driver Build 1.0.4.3008

Following are the fixes and enhancements in this release.

- Fixed an issue of displaying incorrect driver version in UART and arconf after reboot in FreeBSD 12.0.
 - Root Cause: After the outbox driver installation, the inbox driver was not getting blacklisted.
 - Fix: Added the fix in `pkg-install` and `pkg-deinstall` script to load the outbox driver after reboot.
- Added FreeBSD 12 and 11.3 support.

2.2.3.3 Fixes and Enhancements for Solaris Driver Build 1.0.4.3008

There are no known fixes or enhancements for this release.

2.2.3.4 Fixes and Enhancements for Windows Build 106.166.0.1022

Following are the fixes and enhancements in this release.

- Fixed an issue of memory leak in the Normal Tag table.
 - Root Cause: After successfully allocating normal tag table, SmartPQI driver doesn't clear the 'fUsingSpecialTable' flag. This leads to not freeing the Normal Tag Table memory on driver unload.
 - Fix: Set 'fUsingSpecialTable' flag to FALSE upon successful Normal Tag table allocation
- Fixed the Static Driver Verifier (SDV) defects.
 - Root Cause: Driver is not checking pRecord is a NULL pointer or not after getting a record from MapTraverseTrieStartStop function and it catches in SDV.
 - Fix: Forceful check NULL pointer before accessing device records.
- Added a feature of Timeout Support field in pass-through and task management requests.
- Fixed an issue where HLK sleep test are failing with BSOD.
 - Root Cause: `StorportQueueWorkItem` not able to schedule the worker callback immediately after coming back from sleep state. This issue is only visible when the verifier is enabled on `stoport.sys` with "Force IRQL Checking" enabled.
 - Fix: For OS targets dynamically changing `StorportWorkItem` with direct function call during Wake from Sleep/Hibernate. This is a temporary fix and will revert back once Microsoft fixes the issue.
- Fixed an issue where SDDC certification Windows 2016 "PCS-E2ELaunch execute" test case failed with BSOD `irql_not_less_or_equal`.
 - Root Cause: `VoidSrb` is taking null arena memory when releasing `cmdinfo` buffer to arena. After that, immediately the internal controller commands completion is accessing the `cmdinfo srb` and that leads to BSOD `irql_not_less_or_equal`.
 - Fix: Added "else-if" condition to prevent accessing Piggy-backed SRB from command info while processing internal controller commands.

2.2.3.5 Fixes and Enhancements for Windows 7/2008 Build 6.100.0.1014

Following are the fixes and enhancements in this release.

- Fixed an issue where the Windows PNP WHQL tests are failing.
 - Root Cause: Unwanted PQI reset triggering the controller post in SIS mode and due to that all the PNP WHQL test cases failed.
 - Fix: PQI reset is moved out from inappropriate place.
- Fixed an issue where the system freezes during repetition of DC Off/On test.
 - Root Cause: There is a polling routine that may fail if the controller takes too long to respond, causing a potential race condition in `MemAlloc`.

- Fix: Rearranged the order of these host commands and made them all use polling instead of three of them using interrupts.
- Fixed an issue where the dump file (Memory.dmp) gets corrupted upon reboot.
 - Root Cause: During the SmartPQI Dump mode, Power SRB is completed before the flush cache completion. This causes the system to reboot before the cache flush completes and eventually data in the dump file does not get written to the drive.
 - Fix: Shutdown SRB is completed after the successful Flush Cache operation.

2.2.3.6 Fixes and Enhancements for VMware Driver Build 1.0.4.3008

This release provides the following fixes and enhancements:

- Added Timestamp for debugging TMF issues. For debugging TMF related issues, it is better to have the I/O Timestamp printed when OS issues the TMF. This will help to identify how long the I/O really stayed within the driver.
- Fixed an issue of HBA disk hot reinsert failure.
 - Root Cause: SATA HBA disks represent designator type 3h (locally assigned) which VMware does not support to create unique device id.
 - Fix: When HBA mode disks are hot removed, add them to the removed device list instead of removing these devices from the OS SCSI layer and set the disk to an unavailable state, block any subsequent I/O requests to that device until the disk is reinserted. Also, set the disk re-insertion timeout to make sure to remove the devices that are physically removed with no intention of ever reinserting. Upon reinsertion of the disk within the specified timeout period, mark the device state as available and unblock the I/O to the device, then ask SCSI layer to rescan it to bring that device back to online/active state. If disk re-insertion timeout expires that is, disk not re-inserted within the timeout, then try remove the device permanently by informing OS SCSI layer.
- Added a feature of timeout support field in pass-through and task management requests.
 - Details: SCSI pass through commands that are targeted at devices downstream of the controller can get stuck and never complete. Currently, there is no mechanism to recover a pass-through or TMF LUN Reset command that never completes. Firmware added option to specify the timeout in TMF and RAID request IU.
- Added a feature of timeout for driver initiated inquiry.
 - Root Cause: During the device discovery, driver sends inquiry commands to the discovered devices. These requests don't have any timeout specified. This can result in an inquiry never completing, which causes a firmware task to become suspended indefinitely.
 - Fix: Add timeout for internally framed inquiry commands.

2.2.4 Management Software Fixes

2.2.4.1 Fixes and Enhancements for Arconf and MaxView Build B23668

There are no fixes or enhancements in this release.

2.3 Limitations

2.3.1 Firmware Limitations

2.3.1.1 Limitations for Firmware Release 2.62 B0

This release includes the following firmware limitation:

- SATA drives attached to a non-Microsemi expander may get into a failed state when upgrading the controller firmware from previous releases to this release due to the expander not clearing STP affiliation.
 - **Workaround:** Power cycle the expanders to clear the STP affiliation.
- When I/Os are performed on drives that respond slowly or which do not respond to READ or WRITE commands, and when Secure Erase is performed on other SATA drives, I/Os become stalled for a period of time. The time the I/Os are paused depends directly on the amount of unflushed data in the cache and speed with which the device responds to error recovery.
 - **Workaround:** None

2.3.1.2 Limitations for Firmware Release 1.32 Build 0

- Firmware release 1.32b0 may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations.
 - Description: Refer to entry "Fixed an issue where firmware may become unresponsive while attempting to flash firmware or execute other RAID logical volume operations" in the Firmware fixes section.
 - A fix for this issue is available in the 1.60 B0 firmware release. If a firmware flash failure is occurring, try the following workarounds:
 - *Workaround:* If there are no target devices (expanders or drives) attached to the controller, attach a target device to the controller and try the host management operation again.
 - *Workaround:* If the system is operating using UEFI, the HII tool can be used to flash the firmware to this release as outlined in the *Microsemi SmartIOC 2100/SmartROC 3100 Installation and User's Guide (ESC-2170577)*, appendix entry "Updating the SmartIOC 2100/SmartROC 3100 Controller Firmware".
 - *Workaround:* If there are target devices attached to the controller and this issue occurs or none of the workarounds can be used, contact Microsemi Support.

2.3.2 UEFI Limitations

2.3.2.1 Limitations for UEFI Build 1.3.9.2 /Legacy BIOS Build 1.3.9.2

There are no known limitations for this release.

2.3.3 Driver Limitations

2.3.3.1 Limitations for Linux Driver Build 1.2.10-025

This release includes the following limitation:

- Occasionally, the Linux driver might fail to unload. This limitation will occur if maxview services are running in the background and the user tries to unload (disable) the Linux SmartPQI driver with the `rmmmod` command.
 - Workaround: In such a scenario, execute the following commands:
 - `service stor_redfishserver stop`
 - `service stor_tomcat stop`
 - Unload driver

2.3.3.2 Limitations for Windows Driver Build 106.166.0.1022

There are no known limitations for this release.

2.3.3.3 Limitations for Windows 7/2008 Driver Build 6.100.0.1014

There are no known limitations for this release.

2.3.3.4 Limitations for FreeBSD Driver Build 1.0.4.3008

There are no known limitations for this release.

2.3.3.5 Limitations for Solaris Driver Build 1.0.4.3008

This release includes the following Solaris driver limitations:

- UEFI Secure boot is not supported.

2.3.3.6 Limitations for VMware Driver 1.0.4.3008

There are no known limitations for this release.

2.3.4 Hardware Limitations

This release includes the following hardware limitations:

- Two Wire Interface (TWI) address conflicts can cause system DDR memory to not be discovered.
 - *Description:* The HBA1100 boards include two TWI targets on the host-facing SMBUS interface with the following slave addresses:
 - 0xA0 – Field Replaceable Unit (FRU) SEEPROM
 - 0xDE – PBSI (default)
 - *Workaround:* None available. If this issue is encountered, contact your Microsemi support engineer to determine the next steps for your system.
 - *Performance with workaround:* Not applicable
 - *Performance without workaround:* Not applicable
- According to the JEDEC specification, the default TWI addresses for the DDR SPD is 0xA0-0xAE (the spec uses 7 bit addressing which is 0x50-0x57). On platform system board designs with SMBUS wiring that has both PCIe slots and DDR slots shared on the same TWI bus, the TWI devices for the DDR and Smart controller are exposed to address conflicts which can result in the system memory not being discovered. The Smart controller PBSI interface defaults to a value of 0xDE (0x6F in 7-bit addressing) and is not a problem unless it is changed to an address that conflicts with the JEDEC defined values. The Smart controller FRU SEEPROM is hardwired to 0xA0.

2.3.5 Management Software Limitations

2.3.5.1 Limitations for Arconf and maxView Build B23668

There are no known limitations for this release.

3 Updating the Board Firmware for PQI Operation

This section describes how to update the board's firmware components to the latest release.

3.1 Updating Controllers to latest (PQI) Firmware

This procedure describes how to prepare your board to be programmed with the latest board PQI firmware.

Note: Complete these procedures exactly as described for proper functionality. If you do not follow all of the steps correctly, you could encounter unusual runtime behavior.

Flashing the board to the latest PQI firmware:

This section describes how to update all the firmware components on HBA 1100 Adapter boards to the latest release.

If the controller is currently running 1.60 b0 firmware or newer, follow these steps:

1. **Mandatory:** Flash the target with the provided " SmartFWx100.bin" image with arconf/maxView software.
2. **Mandatory:** Cold boot the system to refresh all components.

If the controller is currently running 1.32 b0 firmware, follow these steps:

1. **Mandatory:** Flash the target with the provided "SmartFWx100.bin" image with arconf/maxView software.
 - If the arconf/maxView software becomes unresponsive or hangs then power cycle the system to recover and refer to firmware limitation section [Limitations for Firmware Release 1.32 Build 0](#) on page 15.
2. **Mandatory:** If flashing completes, cold boot the system to refresh all components.

If the controller is currently running 1.04 b0 firmware, follow these steps:

1. **Mandatory:** Flash the controller with the provided "SmartFWx100_v1.29_b314.bin" image with arconf/maxView software.
2. **Mandatory:** Reboot the system to refresh all components.
3. **Mandatory:** Flash the target with the provided " SmartFWx100.bin" image with arconf/maxView software.
4. **Mandatory:** Cold boot the system to refresh all components.

At this point, the controller would be updated and would be ready to use. Install the SmartPQI driver and the latest version of the Arconf/maxView management utility to monitor and configure the controller.

Note: Downgrading firmware could lead to unexpected behavior due to an incompatibility in SEEPROMs between this release and the prior release.

4 Installing the Drivers

See the "Microsemi Adaptec® HBA 1100 Series Host Bus Adapters Installation and User's Guide (ESC-2161232)" for complete driver installation instructions.

**Microsemi Headquarters**

One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113

Outside the USA: +1 (949) 380-6100

Sales: +1 (949) 380-6136

Fax: +1 (949) 215-4996

Email: sales.support@microsemi.com

www.microsemi.com

© 2019 Microsemi. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

ESC-2162192