

=====
Microsemi Adaptec RAID Release Notes dated: January 30, 2017
=====

These release notes contain the following:

1. Description of the Release
2. Supported Controllers
3. Enhancements and Bugfixes

1. Description of the Release:
=====

This is the official software release containing the list of software components listed below:

- Series 8 Firmware Version 7.10.0 Build 33083
- Windows Driver Version 7.5.0.52013
- Linux Driver Version 1.2.1-53005
- VMware Driver Version 1.2.1-53004
- Solaris Version 7.5.0.52025
- FreeBSD Version 7.5.0.52013
- maxView Storage Manager (MSM) Version 2.03 Build 22476

2. Supported Controllers:
=====

- Microsemi Adaptec RAID 8405
- Microsemi Adaptec RAID 8405E
- Microsemi Adaptec RAID 8805
- Microsemi Adaptec RAID 8805E
- Microsemi Adaptec RAID 8885
- Microsemi Adaptec RAID 8885Q
- Microsemi Adaptec RAID 81605Z
- Microsemi Adaptec RAID 81605ZQ

3. Enhancements and Bug Fixes:
=====

General:

- This is a firmware patch release for Series 8 controllers.

Firmware:

Series 8:

- Resolved an issue where the controller would ASSERT if a drive that originally failed out of a RAID 1 was re-inserted while the rebuild was occurring to a hot-spare.
- Resolved an issue where a hot-swap RAID 5 member was exposed to the OS during rebuild.
- Resolved an issue where a drive could be set offline when a drive is hot-plugged during heavy IO.
- Resolved an issue where the parity cache log could cause a controller ASSERT.
- Resolved an issue where the Extended BIOS Data Area (EBDA) was over-sized.
- Resolved a display issue with 6G devices in BIOS.
- Resolved an SSD logical drive creation issue in UEFI.
- Resolved an issue where large firmware images couldn't be updated in UEFI.

Driver:

- Linux/VMware changes:

- Fixed an issue where the OS kernel panicked when removing the AACRAID driver.
 - Root cause: The driver sends wellness commands periodically to the adapter firmware. The commands also contain the driver version as well. When a custom driver is used (as in this case), the driver version size is incremented to beyond what is supported by the command ("custom" string is added to the version). This causes a buffer overflow, and, when the driver is removed, it manifests in OS kernel panic.
 - Fix: Ensure that the driver version size is below the permitted size (use "src" instead of "custom").
 - Exposure: Customers who compile their own versions of the AACRAID drivers and use it with HBA 1000/SmartIOC 2000 cards.
 - Risk: Low
- Fixed an issue where the driver was reporting the adapter dead in case of a SCSI hang.
 - Root cause: When the driver receives an "eh abort" request (SCSI hang) from the kernel, it checks for the status of the adapter. If the adapter's SELF_TEST_FAILED condition is set, the driver returns without resetting the controller. This causes the kernel to offline the adapter.
 - Fix: First check for adapter KERNEL_PANIC before checking the other conditions. This works

- because if a KERNEL_PANIC is detected, the driver resets the adapter.
 - Exposure: All customers.
 - Risk: Low
- Fixed an issue where controller firmware lockup was encountered on the system while deleting units.
 - Root cause: When the driver does not explicitly clear the AdapterProcessed Flag, it causes the firmware to lockup.
 - Fix: Ensure AdapterProcessed is not set when submitting commands.
 - Exposure: All customers using HBA 1000/SmartIO 2000.
 - Risk: Low
- Fixed an issue where customers on Ubuntu/Debian distributions did not have the documentation for building and installing the driver from the source.
 - Root cause: The ARC driver source is distributed in src rpm format to customers. The instructions for compiling src rpm on non-rpm distributions were not provided and non-rpm distributions did not have the required environment setup to use the src rpm files.
 - Fix: The driver source is now also distributed in tar gz format along with instructions to install the driver from source.
 - Exposure: Customers who compile and install the driver (as opposed to those using binary release drivers).
 - Risk: Low
- Added support to include instructions detailing the installation process of DKMS deb driver packages.
 - Root cause: The ARC driver release did not contain documentation on the installation methodology of DKMS deb packages.
 - Fix: Added DKMS deb driver installation documentation.
 - Exposure: Customers using DKMS deb driver packages.
 - Risk: Low
- Fixed an issue where VMware was displaying a PSOD (purple screen) when the driver triggered an IOP reset.
 - Root cause: When the driver issued an IOP reset, it released all the memory and additional resources it had and then reinitialized the driver once the reset went through. When it released the memory, it freed up fib DMA memory at an incorrect misaligned memory location.
 - Fix: Freed the memory at the correct DMA memory location. Fixed an additional memory leak issue.
 - Exposure: Customers using VMware.
 - Risk: Low
- Added sysfs entry to expose the driver version.
 - Root cause: There was no easy way to retrieve the currently running driver version.
 - Fix: Added sysfs entry to expose the driver version.
 - Exposure: Customers who need the currently running driver version without resorting to management tools.
 - Risk: Low
- Fixed an issue where the driver was not recovering a hung controller.
 - Root cause: Driver would get caught in an endless loop resetting the card.
 - Fix: Cleared the Health/Status register before resetting the card. Added state to transition to KERNEL_BOOTING when the firmware starts. Allowed to stay in state until KERNEL_UP_AND_RUNNING is set.
 - Exposure: All customers.
 - Risk: Low
- Removed PCI ID of controllers that are not supported by this driver.
 - Root cause: The current driver claimed support for all the future generation of products, even though those products have their own driver.
 - Fix: Removed PCI ID of future generation cards.
 - Risk: Low
- Fixed driver SCSI hang on "eh reset" request from kernel.
 - Root cause: The driver has two pools of fibs (data transporting abstractions): one for kernel IO and requests, and the other for management commands. The issue was caused when the driver sent kernel IO-tagged fib to the management queue during error recovery.
 - Fix: Added pool tags for fibs to identify and prevent fibs from either pool from being sent to the wrong pool.
 - Exposure: All customers.
 - Risk: Low
- Added support for Oracle Linux (OL) 6.8.
 - Root cause: The driver did not support Oracle Linux 6.8.
 - Fix: Added support for Oracle Linux 6.8.
 - Exposure: Customers who require support for Oracle Linux 6.8.
 - Risk: Low
- Fixed an issue where kdump fails on RHEL 6.5 when it was installed on Microsemi adapter cards.
 - Root cause: Kdump is a Linux mechanism that allows for the capture of kernel logs and cores in case of kernel or driver crash. On CentOS/RHEL systems, kdump is loaded with only a single CPU enabled. When that happens, the Microsemi driver goes into a rudimentary interrupt (intx) mode (which would work in normal cases), but, due to an Intel CPU bug, interrupts are disabled/do not work in kdump mode. This bug prevents the driver from loading and the kernel logs and core from being saved.
 - Fix: To solve this issue, the driver forces full interrupt mode when it detects the driver is in kdump context and the number of CPUs is 1. This enabled the driver to fully load.
 - Exposure: Customers who use RHEL/CentOS and kdump.
 - Risk: Low
- Fixed an issue where customers were not able to install the driver on rpm-based distributions (RHEL/OL/etc.) with custom kernels.
 - Root cause: Currently the ARC driver supports various Linux kernel versions by creating binary drivers for any particular kernel. When a customer uses a custom kernel and if we do not have access to that custom kernel and its headers, it would be difficult to create a driver that supports that kernel. In addition, the customer depends on us for support whenever they upgrade the

- kernel. The alternative is to use a DKMS driver that would support all the custom kernels. Deb DKMS drivers are already supported; need to add support for rpm DKMS drivers.
 - Fix: Added support to generate rpm DKMS drivers.
 - Exposure: Customers who use custom kernels on rpm-based OS distributions.
 - Risk: Low
 - Added support for Oracle Linux UEK3 and UEK4 kernels.
 - Root cause: Currently, the driver supports UEK3 and UEK4 kernels indirectly by supporting the Oracle Linux (OL) distributions that contain either UEK3 or UEK4 kernels. So when a customer needs a particular UEK version driver, they have to use an OL distribution driver version that supports the required UEK kernel version.
 - Fix: Added support to generate OL UEK3 and UEK4 specific binary drivers.
 - Exposure: Customers who require Oracle Linux UEK3 or UEK4 specific drivers.
 - Risk: Low
 - Fixed potential null pointer exception in case of IOP reset failure.
 - Root cause: On IOP RESET, the PCI memory and IO space are freed up, and once the controller is detected to be up and operational, the initialization path is invoked to recreate the IO path resources. If the controller is not operational, the IOP reset path returns with a failure. The command thread invokes check health that relies on the IOP reset mechanism to recover the controller. If the call path returns with an error, however, the driver interprets it incorrectly and continues. It then erroneously pokes a resource pool which may have been freed up by the incomplete reset mechanism in the previous step.
 - Fix: Make sure the driver does not poke the freed resource pool in case of an improper IOP reset.
 - Exposure: All customers.
 - Risk: Low
 - Added support for Oracle Linux (OL) 7.1.
 - Root cause: Driver did not support Oracle Linux 7.1.
 - Fix: Added support for Oracle Linux (OL) 7.1.
 - Exposure: Customers who require support for Oracle Linux 7.1.
 - Risk: Low
 - Added support to ensure proper timeout when the timeout set to 0.
 - Root cause: The kernel at times sends IO commands which at times have a timeout value of 0, which is very short for any drives (HDD or otherwise).
 - Fix: The driver uses a timeout of 30 seconds if it detects timeout as being set to 0.
 - Exposure: All customers.
 - Risk: Low
 - Fixed an issue where the driver eh_reset handler times out while waiting for the driver thread to stop.
 - Root cause: When the adapter crashes or does not respond, the driver tries to send a reset command. In parallel the driver kthread keeps sending periodic sync commands, and because the adapter is not in the correct state, it gets stuck in a fault state. This internally happens when the driver tries to stop its kthread from running but is hung waiting from a response from the adapter.
 - Fix: Allow eh_reset to send a kill signal to the kthread, and then allow eh_reset to start a controller reset.
 - Exposure: All customers.
 - Risk: Low
 - Fixed an issue where drives were being marked offline when IOP reset was performed in the middle of heavy IO.
 - Root cause: The Linux kernel marked drives offline when the adapter crashed in the midst of a heavy IO. These drives did not come back even after the adapter was reset.
 - Fix: Identify any drives that have been marked offline by the kernel and then remove the device. Then invoke a bus rescan in order for the device to show up in the system.
 - Exposure: All customers.
 - Risk: Low
 - Improved the controller reset/recovery mechanism in Linux.
 - Root cause: The driver took 30 minutes to identify whether the adapter has crashed (which is not desirable). The reset logic does not care if the adapter is healthy before issuing an IOP reset, which would cause the request to timeout.
 - Fix: Reduced the health check interval from 30 minutes to one minute. The reset logic includes the health check code in determining the IOP reset. It is necessary to avoid a soft reset mechanism and trigger a hardware reset after giving a one minute delay to allow firmware log capture.
 - Exposure: All customers.
 - Risk: Low
 - Added support for SAS transport in the driver.
 - Root cause: Customers requested support of SAS transport.
 - Fix: Added rudimentary support for SAS transport according to customer requirement. Currently has support to expose the SAS address of the drives and expanders, and to enable them as SAS devices.
 - Exposure: Customers who need SAS transport functionality.
 - Risk: Low
- maxView/ARCCONF:
- Enabled the configuration of the idle session timeout in maxView.
 - Added support for Mode E and Mode F, along with help for expander upgrade in maxView and ARCCONF.
 - Added method name, arguments, and the response data in a format for calls made to CIM in the CIMAdapter.log. Removed unwanted log messages.
 - Added support for the slot information to be displayed for a drive that has a valid secondary location (slot) irrespective of enclosure for Smart Controller.
 - Added code for the following items:
 - Dumping logs in Custompath.
 - Get GOS GUI and agent logs.

- Get particular folder logs.
- Removed additional space from the set properties string in popup message box.
- Upgraded to latest version of JRE to address security issues.
- Modified check to verify if connector functional mode and pending functional mode are not in HBA mode.
- Modified code to allow 128 drives based on ADU report where it is maintained as 128.
- Added changes in storlib and ARCCONF to support Mode 3 support of SATA drive.
- Added a new Boolean "temperatureEventIsGenerated", which will be set to true on the first abnormal temperature event and will be set to false once the temperature reaches normal again. This will allow one event each to be generated whenever the temperature goes from normal to abnormal and vice versa.
- Made the following name changes:
 - Changed "Hard drive access name" to "Disk Name" to sync with the logical drive and physical drive.
 - Added proper validations to display "Disk Name".
- Added a WWID of the SEP device for enclosure of the logical ID instead of from the expander device.
- Added code to capture all debug logs from controller.
- Fixed the following UX issues in the Enclosure log:
 - Fixed Enclosure power supply
 - Fixed Enclosure ID issue
- Added smart controller check for task and uninit commands.
- Fixed an issue where maxView did not detect the controller; fixed the serial number and SES descriptors to fall under ASCII character set.
- Fixed an issue in maxView where the controller could not be found after installing a configured drive when the server was powered off.

 © 2017 Microsemi Corporation. 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.

This software is protected under international copyright laws and treaties. It may only be used in accordance with the terms of its accompanying license agreement.

The information in this document is proprietary and confidential to Microsemi Corporation, and for its customers' internal use. In any event, no part of this document may be reproduced or redistributed in any form without the express written consent of Microsemi, One Enterprise, Aliso Viejo, CA 92656.