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 7.11.0 Build 33173
   - Windows Driver Version 7.5.0.54013
   - Linux Driver Version 1.2.1-54013
   - VMware Driver Version 1.2.1-54013
   - Solaris Version 7.5.0.52025
   - FreeBSD Version 7.5.0.52013
   - maxView Storage Manager (MSM) Version 2.04 Build 22665

2. Supported Controllers:
   ==============================================================
   - Microsemi Adaptec RAID 8005
   - Microsemi Adaptec RAID 8005E
   - Microsemi Adaptec RAID 8805
   - Microsemi Adaptec RAID 8805E
   - Microsemi Adaptec RAID 8885
   - Microsemi Adaptec RAID 81605Z
   - Microsemi Adaptec RAID 81605ZQ

3. Enhancements and Bug Fixes:
   ==============================================================
   General:
   - This is a maintenance release for Series 8 RAID controllers.

   Firmware:
   Series 8:
   - Resolved an issue where the activity LED behavior on CN0/CN1 was different from CN2/CN3.
   - Resolved an issue in NVSRAM that could result in an error.
   - Resolved an issue where unexpected system reboots would occur during Windows 2012 I/O stress testing.
   - Resolved an issue where logical drive creation would fail in some cases when using the Clear method and SATA drives.
   - Resolved an issue where the controller would stop responding during a RAID migration.
   - Resolved an issue where the UART log was flooded with ATA_IDENTIFY_DEVICE and SCSI Inquires on Raw devices.
   - Resolved an issue where Raw drives would not always be recognized within Linux after reboot.
   - Resolved an issue where boot Raw drive was not detected during RedHat installation.
   - Resolved a timing issue during BIOS PCiE configuration that would result in a controller ASSERT.
   - Resolved an issue were a controller ASSERT would occur if a drive is failed during a RAID 1 rebuild.
   - Resolved an issue discovered when a drive is hot-plugged to replace a failed drive.
   - Resolved an issue that would prevent a drive from being initialized or un-initialized.
   - Added controller BIOS Enter and Exit events in controller logs.
   - Resolved an issue where all SATA devices are lost after firmware upgrade in dual-path configurations.

   Driver:
   - Linux/VMware changes:
     - Added support for Ubuntu 14.04.5.
     - Added support for XenServer 7.0.
     - Added support for RHEL 7.3.
     - Added support for SLES 12 SP2.
     - Added support for VMware 6.5.
     - Improved DDKM driver packages.
     - Improved the Ubuntu driver build script.
     - Resolved an issue where driver would cause the system to hang in RHEL 6.5/7.2 under I/O stress testing.
     - Root Cause: When a reset happens, the driver calls scsi_scan_host or aac_sas_scan_host depending on the controller version. This causes the driver to hang.
     - Fix: Fixed the code path that causes the hang.
     - Exposure: Series-8 products running on older Linux kernels 2.6.32 or older
     - Risk: Low
     - Resolved an issue where the system could crash with driver version 1.2.1-53005 and kernel version 3.10.101.
     - Root Cause: The issues were two fold, one the device were all not freed when the driver is removed, next all the devices are not being freed since expanders are allocated as sas_expander and then changed to different class called end device. Now when the driver is removed the kernel skips over the sas_expander class and the kernel complains.
     - Fix: The sas transport is released to ensure that all expanders are assigned to end device type only.
     - Exposure: Series-8 and SmartIOC-2000/HBA-1000 products running Linux kernels 3.10.x such as Debian 8.1, RHEL 7.3, OL 6.8.
     - Risk: Medium
   - Improved controller reset recovery.
     - Root Cause: Several areas were improved to make the controller reset recovery procedure more reliable and responsive.
     - Fix:
       - Reduced the time to initiate the recovery process by polling the controller health more frequently (1 minute).
2. Streamlining of the reset handler to avoid multiple controller resets sent from the driver, in the event of controller hang.
3. Driver logs the controller reset initiation into /var/log/messages.
4. Reduce the burden on the recovering controller, by not sending any sync commands when the controller is on faulty or recovering state.
5. Reduced the time out duration for the sync timeouts, to enable the recovery handlers to kick in a timely manner.

- Resolved an issue where the kernel would not allow the driver to be removed.
  - Root Cause: The reference count of the device exposed by the driver is not zeroed.
  - Fix: Driver ensures that all the allocated memory is freed properly and the reference count is zeroed.
  - Risk: Low
- Resolved an issue where maxView was not displaying the firmware version during controller firmware upgrade.
  - Root Cause: Previous time out value was by default set to 1 sec, which can be too short in cases there are bad behaving devices.
  - Fix: The fix is to identify any drive that have been marked offline by the driver, and remove it from the firmware. Then invoke a bus rescan in order for the device as well as any further change in the configuration to be detected and gets updated in the Linux driver.
  - Risk: Low
- Fixed an issue where maxView was not displaying Tape drive model and negotiated transfer speed.
  - Root Cause: Previously time out value was by default set to 1 sec, which can be too short in cases there are bad behaving devices.
  - Fix: The fix is to identify any drive that have been marked offline by the driver, and remove it from the firmware. Then invoke a bus rescan in order for the device as well as any further change in the configuration to be detected and gets updated in the Linux driver.
  - Risk: Low
- Fixed the Windows 2016 Server hang issue, when devices become unresponsive.
  - Root Cause: When the driver thread performs a periodic time sync and if the firmware is crashed, during that time the driver thread waits for the response that would never arrive from the firmware. This also blocks the controller recovery from happening, as this blocks the reset path from the kernel.
  - Fix: The fix is to allow SIGTERM to terminate the driver thread, resulting in the error handlers to continue the controller recovery.
  - Risk: Low
- Fixed the potential zero memory access if the IOP reset didn’t go through properly.
  - Root Cause: When controller reset, the IOP Memory and I/O space are freed up. If the controller is not operational after the controller reset, there is potential issue for null pointer access in the periodic time sync loop.
  - Fix: The fix is to terminate the periodic loop, if the controller is not restored to a healthy state.
  - Risk: Low

Windows changes:
- Fixed the Windows 2016 Server hang issue, when devices become unresponsive.
  - Root Cause: When the driver thread performs a periodic time sync and if the firmware is crashed, during that time the driver thread waits for the response that would never arrive from the firmware. This also blocks the controller recovery from happening, as this blocks the reset path from the kernel.
  - Fix: Added controller type check.
  - Exposure: Affects Windows drivers for Series-8, Series-7 and Series-6 only.
  - Risk: Low
- After a controller reset, the driver was unable to initialize the adapter.
  - Root Cause: The driver's internal free queue got corrupted. The corruption is due to the driver being allowed to access the driver's 'adapter state' flag before it was reinitialized.
  - Fix: Miniport driver is initializing the internal free queue before setting the 'adapter state' to online.
  - Exposure: Affects Windows drivers for Series-8, Series-7 and Series-6 only.
  - Risk: Low
- Deallocation of zero memory access if the IOP reset didn’t go through properly.
  - Root Cause: When controller reset, the IOP Memory and I/O space are freed up. If the controller is not operational after the controller reset, there is potential issue for null pointer access in the periodic time sync loop.
  - Fix: The fix is to terminate the periodic loop, if the controller is not restored to a healthy state.
  - Risk: Low

maxView/ARCCONF:
- Added support for vSphere 6.5 web client plugin.
- Added support for XenServer 7.0.
- Fixed an issue where maxView was not displaying Tape drive model and negotiated transfer speed.
  - Root cause: ATTR_NAME_DRIVE_MODEL was not available for the tape drive due to an unknown model name.
  - Resolution: Added mapping of product ID to the model so that the model name is shown properly in maxView and ARCCONF.
  - Risk: Medium
- Fixed an issue where maxView was not displaying the firmware version during controller firmware upgrade.
  - Root cause: New firmware version and build info file pointers was changed by the firmware.
  - Resolution: Added the new index and changed the logic for backward compatibility as well.
  - Risk: Low