

# How to flash the Dell PERC H310 (and keep it in the dedicated storage slot)

The Dell PERC H200 and H310 are often built into 11th and 12th generation Dell servers, and are affordable enough to be popular even outside Dell servers. On their stock firmware, these cards behave as HBAs, but also offer some RAID functionalities: the H200 offers basic RAID0, RAID1, RAID10, the H310 also offers RAID5 and RAID50 though it doesn't perform greatly under these additional modes.

The stock firmware can however be limited or problematic in certain instances, i.e. the popular OS FreeNAS does not support accessing SMART data on a card operating with the stock firmware as the megaRAID driver lacks this support; the H310 in particular has a low queue depth, which reduces the performance of the drives, especially if these are able to handle plenty of IOPS, and is therefore unsuitable in some situations, such as usage with vSAN.

In order to work around this, these cards are often flashed with LSI's IT mode firmware, which supports a much greater queue depth (600 vs the original 25 for the H310) and requires a different driver in the OS (mptsas). The mpt driver under FreeBSD supports reading the SMART status.

Dell servers, however, no longer accept the card after it's been flashed to IT mode in the dedicated storage slot, and refuse to boot if such a card is found in it. This can be less of an issue on 11th gen servers, since the H200 uses a standard PCIe socket and there may be others available, but is a crippling limitation for the integrated variant of the H310 used in 12th gen servers. This has led to people having to buy an additional PCIe card and an additional set of miniSAS wires in order to avoid the problem, but is not an optimal solution. Until and unless the server's BIOS is reverse engineered and the check removed, what follows should be used as a better workaround. I recommend to follow this procedure over the traditional flashing done via DOS-mode or UEFI tools since this method should make it harder to put the card in an inconsistent state which will brick it upon reboot.

## TODO

Please report any corrections and suggestions you have to [/u/fourlynx](#) or [fourlynx@phoxden.net](mailto:fourlynx@phoxden.net). I appreciate feedback to improve this guide 🍷

I no longer have 11th generation servers, specifically either an R610 or R710 (which require an H200 Integrated in the dedicated storage slot). As such, I'm in no condition to test whether this guide also works correctly on the H200, although I'm confident it could be easily adapted into that.

I would appreciate contributions in this sense from someone which has those servers, especially since the H200 would NOT brick if presenting the PCI ID of an H200 Adapter.

## Requirements

This tutorial is going to be based around Linux. I recommend to use a live booting image in order to not depend on any drive being present in the server and attached to the card. It should not make any difference which Linux distribution is being used, but I have tested this under Debian stable (stretch).

You will have to install the following packages (change the names as needed to match your distro): `git-core build-essential python3 pciutils p7zip-full`

It is also required to have the ability to pull software from github and to build it. This doesn't have to happen on the live system if for some reason either network connectivity isn't present or if gcc cannot be installed.

You will also need to get the IT-mode firmware to be flashed on the card. This is available through many sources, though I recommend getting it from [LSI-now-Broadcom's website](#).

## Procedure

The first step required is to go in the System setup using F2 during the server startup and to disable the *IOAT DMA Engine* and *SR-IOV Global Enable* inside System BIOS Integrated Devices. These can be enabled again at the end of the procedure.

**System BIOS**

**System BIOS Settings • Integrated Devices**

Integrated RAID Controller  Enabled  Disabled

User Accessible USB Ports

Internal USB Port  On  Off

Embedded NIC1 and NIC2  Enabled  Disabled (OS)

OS Watchdog Timer  Enabled  Disabled

I/OAT DMA Engine  Enabled  Disabled

Embedded Video Controller  Enabled  Disabled

SR-IOV Global Enable  Enabled  Disabled

Memory Mapped I/O above 4GB  Enabled  Disabled

[Slot Disablement](#)

**i** This field enables or disables the integrated RAID controller.

PowerEdge R420      Arrow keys and Enter to select  
 Service Tag : 9P878X1      Esc to exit page, Tab to change focus

**Back**

After rebooting and using the live Linux image, take note of the current SAS address used by the card. We need to rely on MegaCli for this, which is available also for Linux:

```
# wget https://docs.broadcom.com/docs-and-downloads/raid-controllers/raid-controllers-common-files/8-07-14_MegaCLI.zip
# unzip 8-07-14_MegaCLI.zip
# 7z x Linux/MegaCli-8.07.14-1.noarch.rpm
# 7z x MegaCli-8.07.14-1.noarch.cpio
# chmod 755 opt/MegaRAID/MegaCli/MegaCli64
# opt/MegaRAID/MegaCli/MegaCli64 -AdpAllInfo -a0 | grep SAS\ Address
SAS Address      : 5c81f660db770000
```

The SAS address is a 16-byte hexadecimal number that will be restored afterwards. Technically, unless there are multiple HBAs in a single system, all with the same SAS address, it may be left blank (which makes it default to a known value). The MegaCli-8.07.14-1.noarch.rpm package may also be converted to a .deb via the alien utility:

```
# apt install alien
# alien Linux/MegaCli-8.07.14-1.noarch.rpm --scripts
megacli_8.07.14-2_all.deb generated
# dpkg -i megacli_8.07.14-2_all.deb
# /opt/MegaRAID/MegaCli/MegaCli64 -AdpAllInfo -a0 | grep SAS\ Address
```

Next, the following software should be fetched and built:

```
# wget https://github.com/exactassembly/meta-xa-stm/raw/master/recipes-support/lsiutil/files/lsiutil-1.72.tar.gz
# tar xzf lsiutil-1.72.tar.gz
# make -C lsiutil -f Makefile_Linux

# git clone https://github.com/marcan/lsirec
# make -C lsirec
```

Before running the programs just built, it is required to unload the kernel module currently driving the HBA:

```
# rmmod megaraid_sas
```

If drives are attached to the server via the HBA, this will fail.

Hugepages should now be enabled. These will be used to load the IT mode firmware from the host to the card:

```
# echo 16 > /proc/sys/vm/nr_hugepages
```

The card now has to be put in reset mode. For this step, it is required to know its full PCI address. You may use lspci to identify the card:

```
# lspci -nnv | grep H310 -B1
```

```
01:00.0 RAID bus controller [0104]: LSI Logic / Symbios Logic MegaRAID SAS 2008 [Falcon] [1000:0073] (rev 03)
Subsystem: Dell PERC H310 Mini Monolithics [1028:1f51]
```

```
# lsirec/lsirec 0000:01:00.0 unbind
# lsirec/lsirec 0000:01:00.0 halt
```

At this point, the SBR should be extracted from the card. The SBR is a 256-byte blob stored onto a serial EEPROM which contains several pieces of information, most notably how the card should identify itself.

```
# lsirec/lsirec 0000:01:00.0 readsbr H310MM.sbr
```

When flashing to IT mode using the regular utilities, the SBR is erased (typically blanking with the 'megarec.exe -writesbr 0 sbrempty.bin' or the 'sas2flsh.exe -o -e 7' command). This is the step that bricks the card, as the server uses the PCI Subsystem Vendor and Product ID loaded from the SBR to check if the card is allowed or not in the dedicated storage slot. Additionally, the SBR includes checksums, edits should update them as needed or, as we'll do, a tool can be used:

```
# python3 lsirec/sbrtool.py parse H310MM.sbr H310MM.cfg
```

The H310MM.cfg file should have at least two entries modified, PCIPID should be set to 0x0072 and Interface should be set to 0x00 (these two values signal that the card is running in IT/IR mode rather than in megaRAID mode).

#### Note

I've chosen to use the freshly dumped H310MM.cfg as the base for the new SBR, as I've preferred to be minimal in the changes performed, given I do not know the meaning of the other fields. It is however possible (and I have tested the result is working) to use the lsirec/sample\_sbr/sbr\_dell\_h200e\_itir.cfg as the base, modifying the SubsysVID and SubsysPID fields to 0x1028 and 0x1f51 respectively. As long as the card presents the correct SubsysVID and SubsysPID, it will not brick upon reboot. To get the appropriate driver (for megaRAID or IT/IR mode) bound to the card, the PCIPID and Interface fields have to be set to the values described above.

#### Modified H310MM.cfg

```
Unk00 = 0x0022f661
Unk04 = 0xb34f2000
Unk08 = 0x91d700f8
PCIVID = 0x1000
PCIPID = 0x0072
Unk10 = 0x0000
HwConfig = 0x0104
SubsysVID = 0x1028
SubsysPID = 0x1f51
Unk18 = 0x00000000
Unk1c = 0x00000000
Unk20 = 0x00000000
Unk24 = 0x00000000
Unk28 = 0x00000000
Unk2c = 0x00000000
Unk30 = 0x40000000
Unk34 = 0x0300c650
Unk38 = 0x00000000
Unk3c = 0x00000000
Interface = 0x00
Unk41 = 0x2c
Unk42 = 0x0000
Unk44 = 0x00000000
Unk48 = 0x0000
Unk4a = 0x00
```

The modified file should be turned back into a binary blob and written to the card's SBR:

```
# python3 lsirec/sbrtool.py build H310MM.cfg mod_H310MM.sbr
```

```
# lsirec/lsirec 0000:01:00.0 writesbr mod_H310MM.sbr
Device in MPT mode
Using I2C address 0x54
Using EEPROM type 1
Writing SBR...
SBR written from mod_H310MM.sbr
```

You can check that the card's PCI ID has been updated via the following command:

```
lspci -vns 0000:01:00.0 -A intel-conf1 | head -n2
```

The kernel however hasn't noticed this and will be updated later.

Now, you'll want to get the card out from reset mode and instruct it to load the IT mode firmware loaded from the host (rather than the one stored internally)

```
# lsirec/lsirec 0000:01:00.0 hostboot 2118it.bin
```

At this point we can let the kernel know the card has changed, as it is able to operate in IT mode properly as if it had been flashed:

```
# lsirec/lsirec 0000:01:00.0 rescan
```

As the kernel rescans the card, it will also bind the mpt3sas driver to it. This is fine as the next steps, involving actually flashing the firmware to the card to make IT mode persistent, make use of the `/dev/mptctl` device created by the driver.

```
# lsiutil/lsiutil -e
```

### Note

The card may take some time to be brought up into an operational state. If the lsiutil command does not display any controller, rmmod'ing and modprobe'ing the mpt3sas driver might get it working, alternatively, rmmod the mpt3sas driver and repeat the lsirec steps with `unbind halt hostboot rescan`.

Select the adapter, then use *option 46* (Upload FLASH section) and then *option 5* (Complete - all sections) to make a backup. You'll be asked for a file on which to store the dump of the flash for safety.

The card can then be erased via *option 33* (Erase non-volatile adapter storage) and then select in turn *option 3* (FLASH) and *option 8* (Persistent manufacturing config pages). Erasing the flash will take a significant amount of time as it's a sizeable chunk of memory (16 MB). Return to the main menu afterwards.

Now the firmware can be flashed using *option 2* (Download firmware - update the FLASH) and provide the full path to the 2118it.bin file.

To restore the SAS Address of the controller, lsiutil should be used after a reboot via *option 18* (Change SAS WWID). The restored SAS Address will be operative after another reboot (as it can be verified via `systool -c scsi_host -v host0`-- systool is part of the sysfsutils package).

### Note

I could not find a way to reprogram the SAS Address without rebooting the server after flashing the firmware. I have not investigated using the sas2flash tool available for Linux for this task.

```
mpt2sas_cm0: iomem(0x00000000d84c0000), mapped(0xffffb74898be4000), size(16384)
mpt2sas_cm0: ioport(0x0000000000001000), size(256)
mpt2sas_cm0: Allocated physical memory: size(7579 kB)
mpt2sas_cm0: Current Controller Queue Depth(3364),Max Controller Queue Depth(3432)
mpt2sas_cm0: Scatter Gather Elements per IO(128)
mpt2sas_cm0: LSISAS2008: FWVersion(20.00.07.00), ChipRevision(0x03), BiosVersion(00.00
```

### Note

Should you want to boot off a drive attached to the H310MM, you will also have to flash the appropriate bootrom (mptsas2.rom for BIOS, x64sas2.rom for UEFI). I have not included this step as the most popular reason for cards to be flashed is to be used in PCIe passthrough in FreeNAS (or related) VMs. In those configurations, the boot drive will be a vmdk. Not flashing the bootrom also helps speed up the server's boot.

## Credits

Many thanks to [Hector Martin](#) for creating lsirec and writing a first draft documentation on how to operate it to crossflash LSI 2008/2108 HBAs. I recommend to read [this page of his blog](#) to get more info on the SBR and the tools typically used to flash LSI and related, rebranded cards.

I also want to thank @Fohdeesha for listening to months of ramblings on this.