

# Initialize And Finalize Scripts: How do I configure my hardware RAID controller from an initialize script?

*How do I set up a hardware RAID controller from an initialize script?*

An initialize script is used when custom commands need to be executed before checking partitions and mounting devices. For example, to initialize some not-explicitly-supported hardware, or to do a RAID configuration lookup for a particular node. In such cases the appropriate custom commands are added to an initialize script.

Several environment variables are made available to the initialize script. For a complete overview please refer to *Environment Variables Available To initialize And finalize Scripts*, in the [Administrator's manual](#) (Appendix E.3 in version 6.1).

The initialize script might require some executable files e.g. specialized binaries that communicate with the hardware and perform diagnostic or configuration actions.

1. Such files can be placed in a location that will be available to the node-installer such as `/cm/node-installer/opt/<PATH>` on the head node. Then those files and directories will be located in `/opt/<PATH>` in the node-installer's root directory.
2. Alternatively such files can be exported as NFS shares and mounted in the node-installers root.

The initialize script can be configured for a node category or node group. If both the category of a node and the node group of that node have an initialize script set, then the category-level initialize script is executed first.

To set an initialize script for a category in CMSH:

```
[root@ucs-head opt]# cmsht
[ucs-head]% category use usnic
[ucs-head->category[usnic]]% set initializescrpt <PATH>/raid-config.sh

[ucs-head->category[usnic]]% commit
```

*Note that when setting the "initializescrpt" property the contents of the file are stored in CMDaemon's database. Subsequent changes to the file on the disk will not be applied, unless*

# Initialize And Finalize Scripts: How do I configure my hardware RAID controller from an initialize script?

*the "initializescript" property is set again and the updated contents of the file are transferred to the database.*

```
#cat raid-config.sh
```

```
!/bin/bash
```

```
# raid-config.sh
```

```
# This is an example initialize script showing how to configure and LSI raid controller and DELL BIOS
```

```
# using LSI 's MegaCli utility and DELL's pec.
```

```
## Mount NFS share
```

```
mkdir /nfs-mount
```

```
mount <NFS server hostname>:/raid-utils /nfs-mount
```

```
## Location of DELL/LSI utilities
```

```
PEC=/opt/dell/pec
```

```
MEGA=/opt/megacli/MegaCli
```

```
BMCTOOL_IPMITOOL_CMD=/cm/local/apps/ipmitool/current/ipmitool
```

```
# configure the LSI RAID based on the # of drives (currently either 4 or 6)
```

```
NUM_PV=`$MEGA -adpallinfo -aall | awk -F ":" ' / Disks/{print $2}'`
```

# Initialize And Finalize Scripts: How do I configure my hardware RAID controller from an initialize script?

```
NUM_LD=`$MEGA -ldinfo -lall -a0 | grep "Target Id" | wc -l`
```

```
if [ $NUM_LD -ne 2 ]; then          ## if there are already 2 logical volumes, assume the RAID  
has already been configured
```

```
if [ $NUM_PV -eq 4 ]; then
```

```
$MEGA -CfgClr -a0
```

```
$MEGA -CfgLdAdd -r5[252:0, 252:1, 252:2, 252:3] -sz100GB -a0
```

```
$MEGA -CfgLdAdd -r5[252:0, 252:1, 252:2, 252:3] -a0
```

```
fi
```

```
if [ $NUM_PV -eq 6 ]; then
```

```
$MEGA -CfgClr -a0
```

```
$MEGA -CfgLdAdd -r5[252:0, 252:1, 252:2, 252:3, 252:4, 252:5] -sz100GB -a0
```

```
$MEGA -CfgLdAdd -r5[252:0, 252:1, 252:2, 252:3, 252:4, 252:5] -a0
```

```
fi
```

```
fi
```

```
# Configure BIOS and BMC settings based on platform
```

```
PLATFORM=`$PEC/setupbios platform`
```

```
if [ $PLATFORM == 'C6220' ]; then
```

```
$PEC/bmc nic_mode set dedicated
```

```
$PEC/bmc set_chassis_power_cap disable
```

```
$PEC/bmc attr set poweron_stagger_ac_recovery 1
```

```
$PEC/setupbios setting set ioat_dma_engine enabled
```

```
fi
```

# Initialize And Finalize Scripts: How do I configure my hardware RAID controller from an initialize script?

```
if [ $PLATFORM == 'C6100' ]; then
```

```
$PEC/bmc nic_mode set dedicated
```

```
$PEC/bmc attr set poweron_stagger_ac_recovery 1
```

```
$PEC/setupbios setting set hyperthreading_tech enabled
```

```
$PEC/setupbios setting set remote_access enabled
```

```
$PEC/setupbios setting set terminal_type vt_100
```

```
$PEC/setupbios setting set serial_port_number COM2
```

```
fi
```

Unique solution ID: #1152

Author: Panos Labropoulos

Last update: 2013-09-09 18:22