

**SmartRAID 3100 Controller Series SmartHBA 2100 Controller  
Series  
User Guide  
Microsemi MAAS Script and Remote ARCCONF JUJU Charm**

Released  
January 2019



---

a  MICROCHIP company

## Contents

---

<b>1</b>	<b>Revision History .....</b>	<b>1</b>
1.1	Revision 1.0 .....	1
<b>2</b>	<b>Introduction .....</b>	<b>2</b>
<b>3</b>	<b>Overview of MAAS Commissioning Script and Remote ARCCONF Juju Charm .....</b>	<b>3</b>
3.1	MAAS Commissioning Script .....	3
3.2	Remote ARCCONF Juju Charm .....	3
<b>4</b>	<b>Using the MAAS Commissioning Scripts .....</b>	<b>5</b>
4.1	Using MAAS Scripts to Upgrade Controller Firmware .....	5
4.1.1	Firmware Upgrade With Internet Connectivity .....	5
4.1.2	Firmware Upgrade Without Internet Connectivity .....	5
4.1.3	Configuring the Controller Driver on Ubuntu Server 16.04 .....	6
4.2	Using MAAS Scripts to Commission a Node .....	7
<b>5</b>	<b>Using the Remote ARCCONF Juju Charm .....</b>	<b>10</b>
5.1	Installing the Remote ARCCONF Charm .....	10
5.1.1	System Requirements .....	10
5.1.2	Deploying Remote ARCCONF Charm .....	10
5.1.3	Checking Status .....	10
5.1.4	Uninstalling the Remote ARCCONF Charm .....	11
5.2	Remote ARCCONF Charm Use Cases .....	11
5.2.1	Starting Remote ARCCONF .....	11
5.2.2	Remote ARCCONF Commands .....	12

# 1 Revision History

---

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

## 1.1 Revision 1.0

Revision 1.0 was published in November 2018. It was the first publication of this document.

## 2 Introduction

---

This user guide describes how to run Canonical Metal as a service (MAAS) Microsemi commissioning script and explains how to run the Remote ARCCONF utility as a charm using Juju services.

### 3 Overview of MAAS Commissioning Script and Remote ARCCONF Juju Charm

The following sections introduce MAAS and Remote ARCCONF Juju charm.

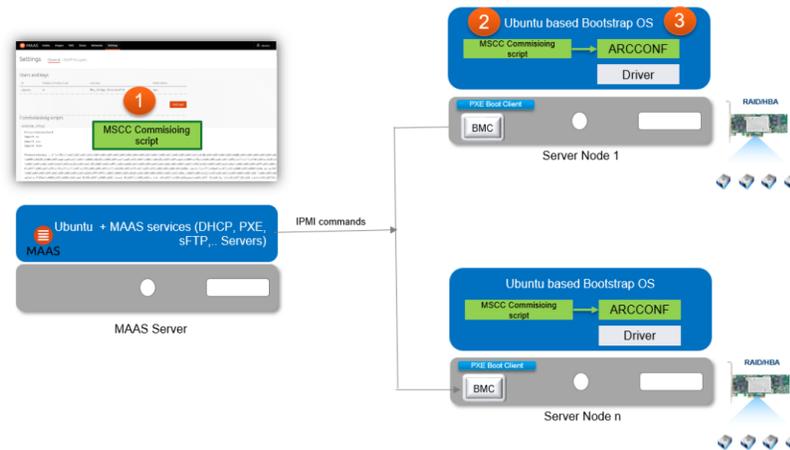
#### 3.1 MAAS Commissioning Script

MAAS allows users to treat physical servers like virtual machines (instances) in the cloud. Rather than managing each server individually, MAAS turns the existing bare metal into an elastic cloud-like resource. MAAS provides management of a large number of physical machines by creating a single resource pool out of them. Participating machines can then be provisioned automatically and used as normal. When those machines are no longer required they are "released" back into the pool.

Microsemi MAAS commissioning scripts allow users to configure Microsemi HBA/RAID controllers during the bare metal provisioning process. Microsemi MAAS commissioning script supports:

- Firmware upgrade
  - Controller: Microsemi HBA and SmartHBA/SmartRAID controllers
  - Expander: SAS Expander Card AEC-82885T
  - Drives connected to the Microsemi controller
- Microsemi HBA/RAID controller configuration can be performed by orchestrating the pre-saved HBA /RAID configuration file, as shown in the following figure.

Figure 1 • Canonical MAAS Commissioning Script

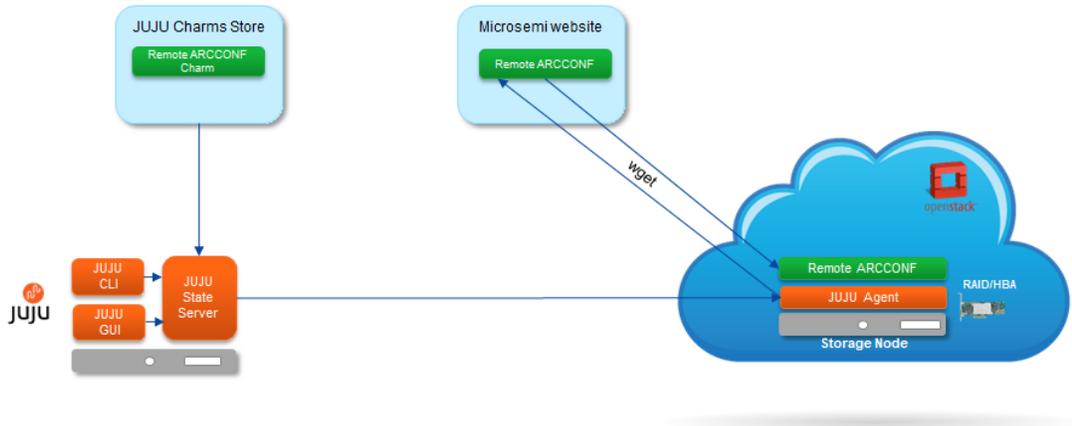


#### 3.2 Remote ARCCONF Juju Charm

Juju is an open-source modelling tool for running software in the cloud. It helps to deploy, configure, manage, maintain, and scale applications quickly and efficiently on public clouds, as well as on physical servers, OpenStack, and containers, as shown in the following figure. The Remote ARCCONF Juju charm allows users to deploy the Remote ARCCONF as a charm in any type of cloud using the Juju GUI or CLI. The smart storage controllers attached to a VMware ESXi hypervisor can be managed through Remote ARCCONF

Disk drives can be grouped into arrays and logical drives using the Remote ARCCONF, . Also, redundancy can be built-in to protect data and to improve system performance.

Figure 2 • Microsemi Juju Charms



## 4 Using the MAAS Commissioning Scripts

MAAS manages a pool of nodes. After registering ("Enlisting" state) a new system and preparing it for service ("Commissioning" state), the system joins the pool and is available for use ("Ready" state). MAAS controls machines through IPMI (or another BMC) or converged chassis controller, such as Cisco UCS.

MAAS users then allocate nodes for their own use ("Acquire") when they go into service. Any subsequently installed operating system will contain the user's SSH public key for remote access (the user's MAAS account first needs to import the key). An allocated MAAS node is *not* like a virtual instance in a cloud: users get complete control of the node, including hardware drivers and root access.

Once a node is no longer needed, it is sent back to the pool for re-use.

For more information about MAAS, see <https://maas.io/>.

### 4.1 Using MAAS Scripts to Upgrade Controller Firmware

Follow the instructions in this section to upgrade the controller firmware on a commissioned node.

#### 4.1.1 Firmware Upgrade With Internet Connectivity

To perform the firmware upgrade on an MSCC smart controller, perform the following steps.

- Provide the packages from the Microsemi website in the URL section with `URL?raw=1`.

**Figure 3 • Firmware Upgrade with Internet**

```
#!/bin/bash -ex
# --- Start MAAS 1.0 script metadata ---
# name: maxView_controller_firmware_upgrade
# title: Controller firmware upgrade
# description: Controller firmware upgrade is used to upgrade the firmware of the controller
# script_type: commissioning
# tags: maxView_controller_firmware_upgrade
# packages:
# url: http://download.adaptec.com/tmp0001/Microsemi_MAAS.tar.gz?raw=1
# may_reboot: False
# --- End MAAS 1.0 script metadata ---

#####
# User configurable parameters
#####

# Enter the SmartPQI driver file name with extension
driverName="smartpqi-dkms_1.1.4.132_all.deb"
# Enter the total number of LUXOR controllers
numberOfController="3"
# Enter the firmware image file name with extension
firmwareImageFileName="SmartFWx100.bin"

#####
# User configurable Ends
#####
```

#### Note

User can edit the file with appropriate URL to flash any specific version of the firmware. By default, the script points to the latest release version of firmware available in the Microsemi website at the time of script release.

#### 4.1.2 Firmware Upgrade Without Internet Connectivity

To upgrade an MSCC smart controller, perform the following steps.

1. Create the tar .gz file containing Remote ARCCONF, smartpqi\* .deb, and firmware image (.bin) files.

**Example:**

To copy all three files to the tar .gz file, change to the directory location and execute the following command.

```
tar -zcvf microsemi_maas_package.tar.gz arccnf smartpqi*.deb
saveconfig.xml
```

2. Copy the files to the following location on the MAAS server.

```
/var/lib/maas/boot-resources/current/filename.tar.gz
```

3. Open the script and update the packages URL attribute with `http://MAAS_IP_ADDRESS:5248/images/filename.tar.gz?raw=1`. The highlighted portion in the following image indicates the content to be updated.

**Figure 4 • Firmware Upgrade without Internet**

```
#!/bin/bash -ex
# --- Start MAAS 1.0 script metadata ---
# name: maxView_controller_firmware_upgrade
# title: Controller firmware upgrade
# description: Controller firmware upgrade is used to upgrade the firmware of the controller
# script_type: commissioning
# tags: maxView_controller_firmware_upgrade
# packages:
# url: http://MAAS_IP:5248/Microsemi_MAAS.tar.gz?raw=1
# may_reboot: False
# --- End MAAS 1.0 script metadata ---

#####
# User configurable parameters
#####

# Enter the SmartPQI driver file name with extension
driverName="smartpqi-dkms_1.1.4.132_all.deb"
# Enter the total number of LUXOR controllers
numberOfController="3"
# Enter the firmware image file name with extension
firmwareImageFileName="SmartFWx100.bin"

#####
# User configurable Ends
#####
```

**Note**

User can edit the file with appropriate URL to flash any specific version of the firmware. By default, the script points to the latest release version of firmware available in the Microsemi website at the time of script release.

### 4.1.3 Configuring the Controller Driver on Ubuntu Server 16.04

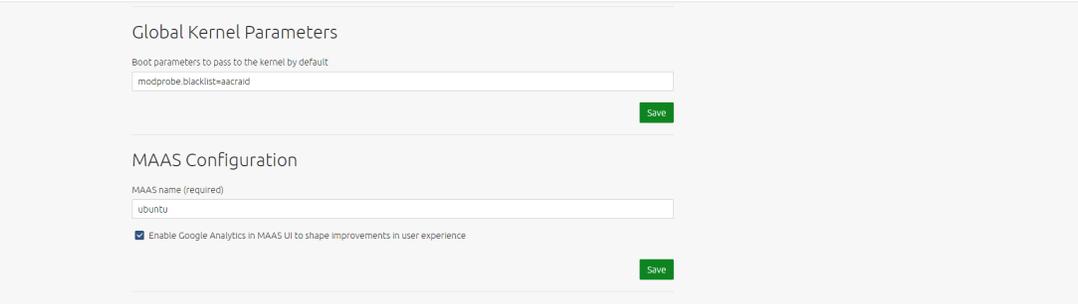
To configure the firmware on the Ubuntu 16.04 server, perform the following steps.

1. Open the script and edit to provide the proper ID for the controller.
 

**Example:** `./arccnf getconfig <controllerID> ld`
2. Go to the MAAS dashboard, under the **Settings** tab, click the **General** tab.

- To blacklist the ARC driver, in the **Global Kernel Parameters** box type `modprobe blacklist=aacraid`, then click **Save**.

**Figure 5 • Global Kernel Parameters**



Settings

**Global Kernel Parameters**

Boot parameters to pass to the kernel by default

`modprobe blacklist=aacraid`

Save

**MAAS Configuration**

MAAS name (required)

ubuntu

Enable Google Analytics in MAAS UI to shape improvements in user experience

Save

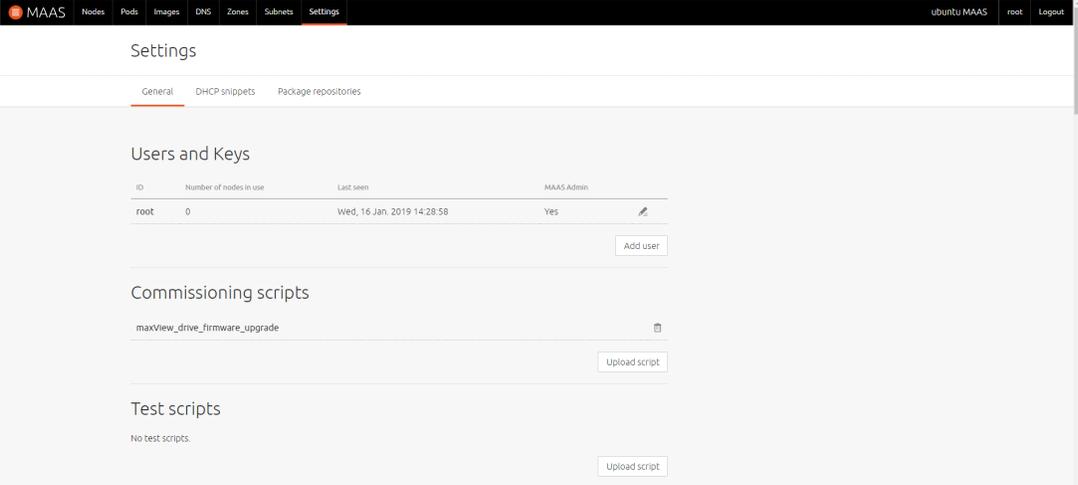
## 4.2 Using MAAS Scripts to Commission a Node

Commissioning scripts are used by MAAS while commissioning and testing a node respectively. Commissioning scripts are used to configure hardware or to perform other tasks during commissioning, such as updating firmware.

To upload the commissioning script, perform the following steps.

- Go to the MAAS dashboard at <http://10.187.66.68:5240/MAAS/#/dashboard>, then click the **Setting** tab.
- To upload an appropriate script, under the **Commissioning scripts** section under **General** tab, click **Upload Script**.

**Figure 6 • Upload User Script**



MAAS Nodes Pods Images DNS Zones Subnets Settings ubuntu MAAS root Logout

Settings

General DHCP snippets Package repositories

**Users and Keys**

ID	Number of nodes in use	Last seen	MAAS Admin
root	0	Wed, 16 Jan. 2019 14:28:58	Yes

Add user

**Commissioning scripts**

maxview\_drive\_firmware\_upgrade

Upload script

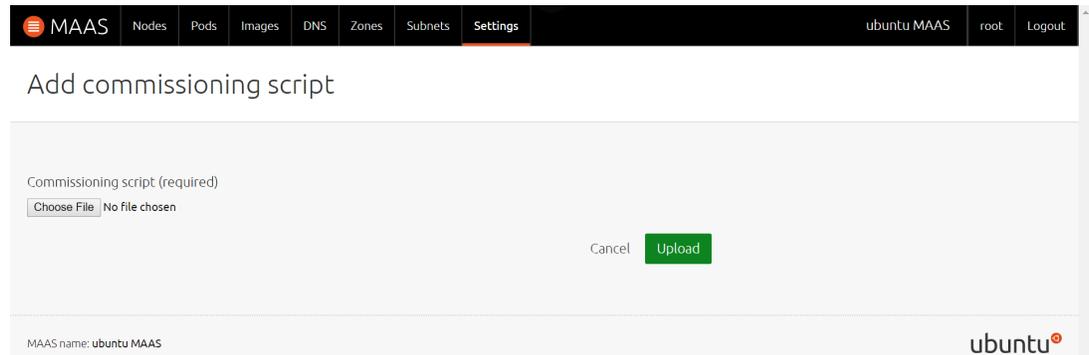
**Test scripts**

No test scripts.

Upload script

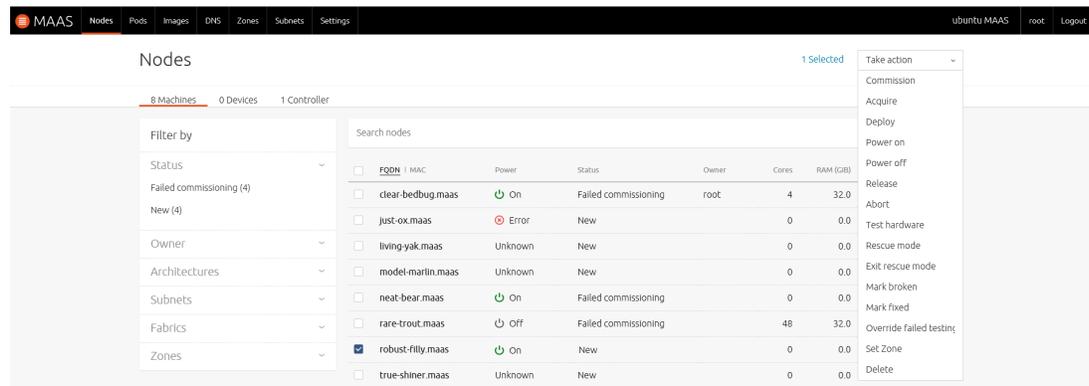
- In the **Add commissioning script** page, to browse to the appropriate script, click **Choose File**, and then click **Upload**. The uploaded script file will be listed under the **Commissioning scripts** section in the **Settings** page.

**Figure 7 • Upload User Script**



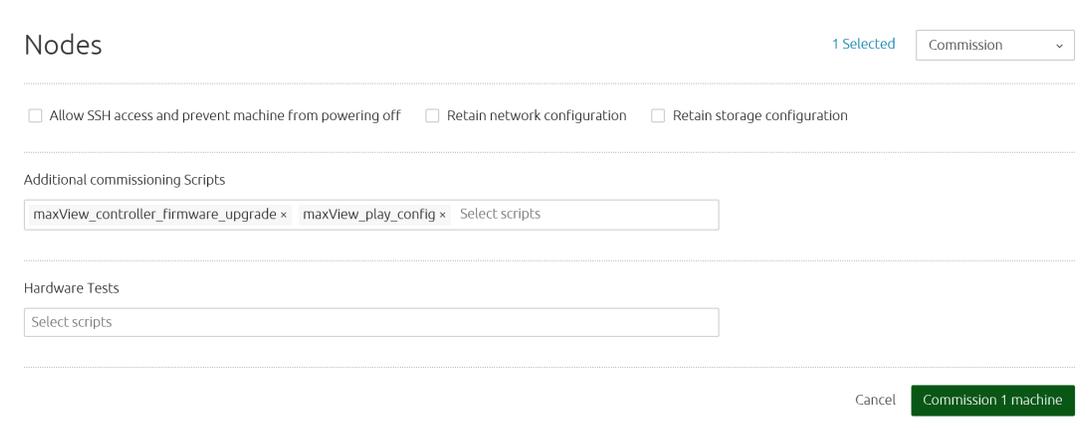
- To commission a node with the required scripts, click the **Nodes** tab. The **Nodes** page displays a list of available nodes. Select an appropriate node, Click **Take Action**, and then click **Commission**.

**Figure 8 • Select Node to Commission**



- In the **Nodes** page, add or remove scripts in the **Additional commissioning Scripts** field and click **Commission 1 Machine**.

**Figure 9 • Commission a Node**



6. To view results of the operation on the selected node, click the **View log** link. The **Output** page appears.
7. On the **Output** page, click **stdout** to view the firmware upgrade status.

## 5 Using the Remote ARCCONF Juju Charm

The following sections describe how to work with Remote ARCCONF Juju charm.

### 5.1 Installing the Remote ARCCONF Charm

The following sections describe how to install the Remote ARCCONF Juju charm.

#### 5.1.1 System Requirements

To install Remote ARCCONF charm, a central Juju controller (machine) is required. The online hosted Juju controller, that is, Juju as a service (JAAS), can also be used for the purpose.

#### Note

To test locally, configure a Juju controller on the local machine. For the local instance of the Juju controller, an Ubuntu 16.04 system is required. For more information, see <https://docs.jujucharms.com/2.4/en/tut-1xd>.

#### Note

Remote ARCCONF support is available for only VMware ESXi hypervisor.

#### 5.1.2 Deploying Remote ARCCONF Charm

To deploy the Remote ARCCONF charm from the charm store, execute the following command.

```
juju deploy cs:sddc.support/remotearconf-1 --series trusty
```

#### 5.1.3 Checking Status

To check the status, execute the following command.

```
Juju status
```

The following figure shows a sample output.

Figure 10 • Juju Status

```
ubuntu@ubuntu:~$ juju status
Model      Controller  Cloud/Region  Version
default    test        localhost/localhost  2.1.2

App        Notes      Version  Status      Scale  Charm          Store      Rev  OS
remotearconf  maintenance  1  remotearconf  jujucharms  4  ubun

Unit      Workload      Agent  Machine  Public address  Ports  Message
remotearconf/6*  maintenance  idle   20       10.106.77.114   Creating r
emote arconf configuration

Machine  State  DNS           Inst id          Series  AZ
20       started  10.106.77.114  juju-36a255-20  trusty

ubuntu@ubuntu:~$
```

### 5.1.4 Uninstalling the Remote ARCCONF Charm

To uninstall the Remote ARCCONF charm, the application needs to be removed from the model.

Execute the following command to uninstall Remote ARCCONF.

```
juju remove-application remotearcconf
```

## 5.2 Remote ARCCONF Charm Use Cases

This section introduces the main features of Remote ARCCONF charm. It also explains how to get help for various commands.

### 5.2.1 Starting Remote ARCCONF

To start Remote ARCCONF, perform the following steps.

1. Execute the following command in the bash shell for lxc container (local) deployed charm.

```
lxc exec <instance_id> /bin/bash
```

The following figure shows the output of the command.

**Figure 11 • Run Juju**

```
ubuntu@ubuntu:~$ sudo lxc exec juju-36a255-20 /bin/bash
[sudo] password for ubuntu:
root@juju-36a255-20:~#
```

#### Note

Users need root privilege to run Remote ARCCONF. Otherwise, they must provide the root password for the system.

2. To access the application at `/opt/remotearcconf`, change the directory to the corresponding folder by executing the following command.

```
cd /opt/remotearcconf
```

The following figure shows the output of the command.

**Figure 12 • Enter Working Directory**

```
root@juju-36a255-20:~# cd /opt/remotearcconf
root@juju-36a255-20:/opt/remotearcconf#
```

3. Follow the instruction in `Install.txt` and run the following command to set the `LD_LIBRARY_PATH`.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD
```

The following figure shows the output of the command.

**Figure 13 • Set LD\_LIBRARY\_PATH**

```
root@juju-36a255-20:/opt/remotearcconf# export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD
root@juju-36a255-20:/opt/remotearcconf#
```

4. Connect the VMware ESXi hypervisor system where the controller is connected by executing the following command.

```
./arcconf setvmcredential <ESXIP> <ESXCIMOMPORT> <ESXUSERID> <ESXPASSWORD>
```

Where,

- `ESXIP`: The ip address of VMware esxi hypervisor machine.
- `ESXCIMOMPORT`: This is the CIMOM Port on ESXi machine.
- `ESXUSERID`: This is the user id on ESXi machine.
- `ESXPASSWORD`: This is the password of ESXi machine.

After executing this command, the controller is ready to be managed.

## 5.2.2 Remote ARCCONF Commands

To see the list of available commands, execute `arcconf` at the bash prompt without any parameters.

Figure 14 • ARCCONF Commands

```

root@juju-36a255-20:/opt/remotearccconf# ./arccconf
Controllers found: 1

| UCLI | Microsemi Adaptec uniform command line interface
| UCLI | Version 3.00 (B23458)
| UCLI | (C) Microsemi Corporation 2003-2018
| UCLI | All Rights Reserved

ATAPASSWORD          | setting password on a physical drive
CONSISTENCYCHECK    | toggles the controller background consistency check mode
CREATE               | creates a logical device
DELETE              | deletes one or more logical devices
EXPANDERLIST        | lists the expanders connected to the controller
EXPANDERUPGRADE     | updates expander firmware
GETCONFIG           | prints controller information
GETLOGS            | gets controller log information
GETSMARTSTATS      | gets the SMART statistics
GETSTATUS           | displays the status of running tasks
GETVERSION         | prints version information for all controllers
IDENTIFY           | blinks LEDs on device(s) connected to a controller
IMAGEUPDATE        | update physical device firmware
KEY                 | installs a Feature Key onto a controller
LIST                | lists all controllers connected to the system
MODIFY             | performs RAID Level Migration or Online Capacity Expansion
PHYERRORLOG        | displays PHY error logs for controller or device or an expander PHY
PLAYCONFIG         | apply the configurations on controller(s) from input XML
RESCAN             | checks for new or removed drives
RESETSTATISTICSCOUNTERS | resets the controller statistics counters
ROMUPDATE          | updates controller firmware
SAVECONFIG         | saves the controller(s) information XML file
SAVESUPPORTARCHIVE | saves the support archive
SETARRAYPARAM     | sets the parameters of an array
SETBOOT           | marks a device bootable
SETCACHE          | adjusts physical or logical device cache mode
SETCONFIG         | restores the default configuration
SETCONNECTORMODE  | changes connector mode settings
SETCONTROLLERMODE | changes Controller mode settings
SETCONTROLLERPARAM | sets the parameters of the controller
SETMAXCACHE       | adjusts maxCache settings for physical or logical device
SETNAME          | renames a logical device given its logical device number
SETPERFORM       | changes adapter settings based on application
SETPOWER         | power settings for controller or logical device
SETPRIORITY      | changes specific or global task priority
SETSTATE        | manually sets the state of a physical or logical device
SETSTATSDATACOLLECTION | toggles the controller statistics data collection mode
SETVMCREDENTIAL  | Storing the ESX server credentials in Guest OS in encrypted format
SLOTCONFIG       | lists devices attached to each Slot in an Enclosure
SMP              | sends SMP Commands to Expander
SPLITMIRROR     | manages splitting and backup of a mirror
TASK            | performs a task which is applicable on a physical or logical device
UNINIT          | manually uninitialized the physical devices which are raw or ready

```

If the command fails, immediately an error message for the failed command is displayed.

For more details on the CLI commands, see *ARCCONF Command Line Utility User Guide for Microsemi Smart Storage Controllers (ESC-2161616)*.

**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](mailto:sales.support@microsemi.com)

[www.microsemi.com](http://www.microsemi.com)

© 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](http://www.microsemi.com).

ESC-2182129