

**Adaptec Smart Storage Controllers**  
**User's Guide**  
**ARCCONF Command Line Utility**

Released  
October 2020



---

a  **MICROCHIP** company

## Revision History

Revision	Revision Date	Details of Change
9.0	October 2020	<p>Revision 9.0 is a post-production release of this document published in October 2020. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• Added Appendix for Controller, Array, Logical, and Physical Device properties</li> <li>• arconf create - Updated parameters, Stripesize and SSD I/O BYPASS</li> <li>• arconf getconfig - Updated description</li> <li>• arconf SETCONTROLLERPARAM - Updated syntax, parameter and examples; included DISCOVERYPROTOCOL and EXPANDERSCANDURATION</li> <li>• arconf task - Updated the description for Random Zero and Random Random Zero in 'options' parameter</li> <li>• Updated syntax of arconf commands to include an optional parameter "[nologs]"</li> </ul>
8.0	February 2020	<p>Revision 8.0 is a post-production release of this document published in February 2020. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• arconf setcontrollerparam - Updated the syntax, parameter and examples of MCTP/PBSI help</li> </ul>
7.0	October 2019	<p>Revision 7.0 is a post-production release of this document published in October 2019. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• arconf romupdate- Updated parameter and examples</li> <li>• arconf create- Updated Parameters section</li> <li>• arconf getstatus- Updated note</li> <li>• arconf getversion- Updated description</li> <li>• arconf setperform- Updated syntax, parameters and examples sections</li> </ul>
6.0	February 2019	<p>Revision 6.0 is a post-production release of this document published in March 2019. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• arconf getlogs-Updated syntax, parameters and example</li> </ul>
5.0	January 2019	<p>Revision 5.0 is a post-production release of this document published in October 2018. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• Added commands:</li> <li>• arconf consistencycheck- Added PARALLELCOUNT, EVENTNOTIFY, INCONSISTENCYREPAIRPOLICY parameters</li> <li>• arconf create-Updated Controller and Array parameters</li> <li>• arconf delete-Added maxCache Id parameter and updated ALL</li> <li>• arconf expanderupgrade-Updated Mode parameter</li> <li>• arconf getconfig-Added LD and AR parameters</li> <li>• arconf identify-Added MAXCACHE parameter</li> <li>• arconf imageupdate-Updated Mode parameter</li> <li>• arconf modify-Added EXPAND and SHRINK parameter</li> <li>• arconf phyerrorlog-Updated example</li> <li>• arconf romupdate-Added ImagePath, toggle and once parameters</li> <li>• arconf savesupportarchive-Added parameters GUI and Redfish</li> <li>• arconf setcache-Added DRIVEWRITECACHEPOLICY, WRITECACHEBYPASSTHRESHOLD and RECOVERCACHEMODULE parameters</li> <li>• arconf setcontrollerparam-Updated I2CADDRESS parameter</li> <li>• arconf setmaxcache-Added MAXCACHEWRITEPOLICY</li> <li>• arconf setstate-Added maxCache Id and SPARETYPE</li> <li>• arconf Task-Added start array and start logical drive</li> </ul>
4.0	April 2018	<p>Revision 4.0 is a post-production release of this document published in March 2018. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• Add commands: maxcrypto, maxcryptoaccounts, maxcryptokey</li> <li>• arconf Create-Add ENCODE parameter</li> <li>• arconf SetControllerParam-Add MIXEDVOLUMES, FWLOCK parameters</li> <li>• arconf Task-Add ENCODE, CRYPTOERASE, REKEY parameters</li> <li>• arconf SetConfig-Add CLEARMAXCRYPTOCONFIG parameter</li> <li>• arconf GetConfig-Display maxCrypto properties</li> <li>• Add section: Running ARCCONF from the UEFI Shell</li> <li>• Add commands supported in UEFI/ARCCONF only: passthrough, slotconfig</li> </ul>
3.0	September 2017	<p>Revision 3.0 is a post-production release of this document published in September 2017. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• arconf getconfig-Added logical drive maxCache statistics to the list of information provided by this command.</li> </ul>

Revision	Revision Date	Details of Change
		<ul style="list-style-type: none"> <li>• arccnf setarrayparam-Added SSDIOBYPASS to parameter.</li> <li>• arccnf setcontrollerparam-Added SANITIZELOCK parameter.</li> <li>• arccnf task-Added CHANNEL# ID# parameter and options for secureerase.</li> <li>• arccnf uart-Removed this command.</li> </ul>
2.0	February 2017	<p>Revision 2.0 is a post-production release of this document published in February 2017. The following is a summary of the changes:</p> <ul style="list-style-type: none"> <li>• Downloading the Installation Packages-Removed ".exe" from the Linux command.</li> <li>• arccnf create-Added maxCache support and added RAID levels. Added note to avoid mixing SMR and PMR in an array.</li> <li>• arccnf getconfig-Added display of controller manufacturing information, green backup, associated split mirror array information, and I2C address, clock speed and clock stretching information.</li> <li>• arccnf identify-Added display of logical drive and array.</li> <li>• arccnf modify-Added note to avoid mixing SMR and PMR in an array.</li> <li>• arccnf romupdate-Updated download URL for .bin file.</li> <li>• arccnf setcontrollerparam- Added I2CADDRESS.</li> <li>• arccnf setmaxcache-Added this new command.</li> <li>• arccnf setpower-Added this new command.</li> </ul>
1	August 2016	Preliminary Release for Early Customer Engagement.

## Software License Agreement

---

PLEASE READ CAREFULLY: THE USE OF THIS SOFTWARE IS SUBJECT TO THE SOFTWARE LICENSE TERMS OF MICROSEMI, INC. AND OTHER LICENSORS WHOSE SOFTWARE MAY BE BUNDLED WITH THIS PRODUCT.

BY YOUR USE OF THE SOFTWARE INCLUDED WITH THIS PRODUCT YOU AGREE TO THE LICENSE TERMS REQUIRED BY THE LICENSOR OF THAT SOFTWARE, AS SET FORTH DURING THE INSTALLATION PROCESS. IF YOU DO NOT AGREE TO THE LICENSE TERMS APPLICABLE TO THE SOFTWARE, YOU MAY RETURN THE ENTIRE UNUSED PRODUCT FOR A FULL REFUND.

In return for acquiring a license to use the Microsemi software, which may include software from third party licensors and patches made available by Microsemi (“Software”), and the related documentation, you agree to the following terms and conditions:

1. License. This Agreement grants you, the Licensee, a license to:
  - a. Use the Software on a single computer system, which is not intended for use by more than five (5) users; and:
  - b. Make one copy of the Software in machine readable form solely for back-up purposes, provided you reproduce Microsemi's copyright proprietary legends. Notwithstanding the foregoing, the Software may be used on the home, laptop or other secondary computer of the principal user of the Software, and an additional copy of the Software may be made to support such use. As used in this license, the Software is “in use” when it is either loaded into RAM or installed on a hard disk or other permanent memory device. The Software may be “in use” on only one computer at any given time. (Different license terms and fees are applicable for networked or multiple user applications.) As a specific condition of this license, you agree to use the Software in compliance with all applicable laws, including copyright laws, and that you will not copy, transmit, perform or distribute any audio or other content using the Software without obtaining all necessary licenses or permissions from the owner of the content.
2. Restrictions. You may not distribute copies of the Software to others or electronically transfer the Software from one computer to another over a network. You may not post or otherwise make available the Software, or any portion thereof, in any form, on the Internet. You may not use the Software in a computer service business, including in time sharing applications. The Software contains trade secrets and, in order to protect them, you may not decompile, reverse engineer, disassemble, or otherwise reduce the Software to a human-perceivable form. YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, LOAN, RESELL FOR PROFIT, DISTRIBUTE, NETWORK OR CREATE DERIVATIVE WORKS BASED UPON THE SOFTWARE OR ANY PART THEREOF.
3. Ownership of Software. As Licensee, you own the media upon which the software is recorded or fixed, but Microsemi and its licensors retain title and ownership of the Software recorded on the original media and all subsequent copies of the Software, regardless of the form or media in which or on which the original and other copies may exist. This license is not a sale of the Software or any copy.
4. Confidentiality. You agree to maintain the Software in confidence and that you will not disclose the Software to any third party without the express written consent of Microsemi. You further agree to take all reasonable precautions to preclude access of unauthorized persons to the Software.
5. Term. This license is effective until January 1, 2045, unless terminated earlier. You may terminate the license at any time by destroying the Software (including the related documentation) together with all copies or modifications in any form. Microsemi will have the right to terminate our license immediately if you fail to comply with any term or condition of this Agreement. Upon any termination, including termination by you, you must destroy the Software (including the related documentation), together with all copies or modifications in any form.
6. Special Terms Applicable to Databases. Where a database is included with the Software, you acknowledge that it is licensed only in connection with the use of the Software to perform disc creation, and that the

database and all data derived therefrom must be maintained in confidence in accordance with the provisions of Section 4. This license does not grant you any rights to distribute or disclose such database or data.

- 7. Limited Warranty.** Microsemi and its Licensor warrant only that the media upon which the Software is furnished will be free from defects in material or workmanship under normal use and service for a period of thirty (30) days from the date of delivery to you. MICROSEMI AND ITS LICENSORS DO NOT AND CANNOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THE SOFTWARE OR DOCUMENTATION. THE FOREGOING STATES THE SOLE AND EXCLUSIVE REMEDIES MICROSEMI AND ITS LICENSORS WILL PROVIDE FOR BREACH OF WARRANTY. EXCEPT FOR THE FOREGOING LIMITED WARRANTY, MICROSEMI AND ITS LICENSORS MAKE NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED, AS TO NON-INFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not apply to you. This warranty gives you specific legal rights and you may also have other rights which vary from state to state.
- 8. The entire liability of Microsemi and its licensors, and your exclusive remedy for a breach of this warranty, shall be:**

  - a.** The replacement of any media not meeting the above limited warranty which is returned to Microsemi; or:
  - b.** if Microsemi or its distributor is unable to deliver replacement media which is free from defects in materials or workmanship, you may terminate this Agreement by returning the Software and your money will be refunded.
- 9. Limitation of Liability.** IN NO EVENT WILL MICROSEMI OR ITS LICENSORS BE LIABLE TO YOU FOR ANY INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR LOSS OF DATA, EVEN IF MICROSEMI OR A LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Some states do not allow the exclusion or limitation of special, incidental, or consequential damages, so the above limitation or exclusion may not apply to you.
- 10. Export.** You acknowledge that the laws and regulations of the United States and other countries may restrict the export and re-export of the Software. You agree that you will not export or re-export the Software or documentation in any form in violation of applicable United States and foreign law.
- 11. Government Restricted Rights.** The Software is subject to restricted rights as follows. If the Software is acquired under the terms of a GSA contract: use, reproduction or disclosure is subject to the restrictions set forth in the applicable ADP Schedule contract. If the Software is acquired under the terms of a DoD or civilian agency contract, use, duplication or disclosure by the Government is subject to the restrictions of this Agreement in accordance with 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors and 49 C.F.R. 227.7202-1 of the DoD FAR Supplement and its successors.
- 12. General.** You acknowledge that you have read this Agreement, understand it, and that by using the Software you agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between Microsemi and you, and supersedes any proposal or prior agreement, oral or written, and any other communication between Microsemi and you relating to the subject matter of this Agreement. No additional or any different terms will be enforceable against Microsemi unless Microsemi gives its express consent, including an express waiver of the terms of this Agreement, in writing signed by an officer of Microsemi. You assume full responsibility for the use of the Software and agree to use the Software legally and responsibly. This Agreement shall be governed by California law, except as to copyright matters, which are covered by Federal law. This Agreement is deemed entered into at Sunnyvale, California by both parties. Should any provision of this Agreement be declared unenforceable in any jurisdiction, then such provision shall be deemed severable from this Agreement and shall not affect the remainder hereof. All rights in the Software not specifically granted in this Agreement are reserved by Microsemi.

Should you have any questions concerning this license, contact:



Microsemi Corporation  
Legal Department  
One Enterprise  
Aliso Viejo, CA 92656  
(949) 380-6100

# Contents

---

<b>1 Getting Started with the Command Line Utility.....</b>	<b>1</b>
1.1 Installing the Command Line Utility.....	1
1.1.1 Downloading the Installation Packages.....	1
1.1.2 Installing Remote ARCCONF.....	1
1.2 Starting the Command Line Utility.....	2
<b>2 Using the Command Line Utility.....</b>	<b>3</b>
2.1 ARCCONF Commands.....	4
2.2 arconf atpassword.....	5
2.3 arconf consistencycheck.....	5
2.4 arconf create.....	6
2.5 arconf delete.....	8
2.6 arconf driverupdate.....	9
2.7 arconf expanderlist.....	9
2.8 arconf expanderupgrade.....	10
2.9 arconf getconfig.....	11
2.10 arconf getlogs.....	13
2.11 arconf getsmartstats.....	13
2.12 arconf getstatus.....	14
2.13 arconf getversion.....	14
2.14 arconf identify.....	15
2.15 arconf imageupdate.....	16
2.16 arconf key.....	16
2.17 arconf list.....	17
2.18 arconf maxcrypto.....	17
2.19 arconf maxcryptoaccounts.....	19
2.20 arconf maxcryptokey.....	20
2.21 arconf modify.....	21
2.22 arconf passthrough.....	22
2.23 arconf phyerrorlog.....	25
2.24 arconf playconfig.....	25
2.25 arconf rescan.....	26
2.26 arconf resetstatisticscounters.....	27
2.27 arconf romupdate.....	27
2.28 arconf saveconfig.....	28
2.29 arconf savesupportarchive.....	29
2.30 arconf setarrayparam.....	30
2.31 arconf setboot.....	30
2.32 arconf setcache.....	31
2.33 arconf setconfig.....	32
2.34 arconf setconnectormode.....	33
2.35 arconf setcontrollermode.....	34
2.36 arconf setcontrollerparam.....	34
2.37 arconf setmaxcache.....	37
2.38 arconf setname.....	38
2.39 arconf setperform.....	39
2.40 arconf setpower.....	39
2.41 arconf setpriority.....	40
2.42 arconf setstate.....	40

2.43 arconf setstatsdatacollection.....	42
2.44 arconf slotconfig.....	42
2.45 arconf smp.....	43
2.46 arconf splitmirror.....	44
2.47 arconf task.....	45
2.48 arconf uninit.....	46
<b>Appendix A Running ARCCONF in the UEFI Shell.....</b>	<b>48</b>
<b>Appendix B Controller, Array, Logical and Physical Device Properties.....</b>	<b>51</b>



# 1 Getting Started with the Command Line Utility

---

This guide explains how your Microsemi Smart Storage controller supports the use of the ARCCONF command line utility.

This utility allows you to:

- Create and delete logical drives
- Encrypt and decrypt logical drive data (if supported by your controller)
- Display configuration settings
- Copy configurations from one computer to another
- Flash new firmware and BIOS onto the controller
- Enable the controller to check the removal and connection of any disk drives
- Provide access to the status and event logs of a controller

**Note:** This guide focuses on using ARCCONF with Adaptec Smart Storage Controllers (SmartRAID/SmartHBA/SmartIOC/SmartROC). For information about using ARCCONF with Microsemi Adaptec Series 8 (legacy) RAID controllers, see the *Microsemi Adaptec RAID Controller Command Line Utility User's Guide* (ESC-2160659).

## 1.1 Installing the Command Line Utility

Follow the instructions in this section to install ARCCONF on the supported operating systems.

### 1.1.1 Downloading the Installation Packages

Complete these steps to download the ARCCONF installation package for your operating system(s):

1. Open a browser window, then type `start.microsemi.com` in the address bar.
2. Navigate to your controller product page, then select Storage Manager downloads.
3. Download the ARCCONF Command Line Utility installation package.
4. When the download completes, extract the package contents to the installation directory on your machine (`Program Files` or `/opt`, for instance).
5. On Linux systems, ensure that `arccconf` has 'execute' privilege:  

```
chmod arccconf +x
```

### 1.1.2 Installing Remote ARCCONF

Use the following procedure to install Remote ARCCONF on a VMware ESXi system. Remote ARCCONF provides command line support on Windows and Linux Guest OSs.

1. Copy the `arccconf` folder to the remote machine using the Remote Desktop Connection utility (on Windows) or a remote copy utility, such as `putty` or `scp` (on Linux).

**Note:**

You can also get remote `arccconf` from the `maxView™` installation directory in the `esx_arccconf` folder.

2. Run `arccconf` from the installation directory.

## 1.2 Starting the Command Line Utility

**Note:** You can run a subset of ARCCONF commands from the UEFI shell. For more information, see [Running ARCCONF in the UEFI Shell](#) on page 48.

1. To start ARCCONF, enter one of the following commands:

Option	Description
<b>Windows</b>	<install_dir>\arconf.exe
<b>Linux</b>	/<install_dir>/arconf
<b>VMware ESXi with Remote ARCCONF</b>	/usr/RemoteArconf/arconf

where Install\_dir is the directory where the utility is installed.

2. To see a list of available commands, type ARCCONF at the prompt. For help with a specific command, type ARCCONF <command\_name> help.

## 2 Using the Command Line Utility

---

This chapter explains how to use the command line utility interactively or in batch mode. With interactive mode, enter commands at the prompt. In batch mode, create scripts and run the script in the appropriate shell, as described in the following table:

**Table 1 • ARCCONF Batch Environments**

Environment	Batch File	Run Script
Windows	.bat	CMD.EXE
Linux/Unix	.sh	sh / bash

In either mode, if your command fails, you immediately see an error message of command failed. Other script messages that you can get are command completed successfully, or command aborted.

The return values for each command are the same:

```
0x00: SUCCESS
```

```
0x01: FAILURE - The requested command failed
```

```
0x02: ABORT - The command was aborted because parameters failed validation
```

```
0x03: INVALID_ARGUMENTS - The arguments are incorrect. (Displays COMMAND help)
```

```
0x06: INVALID_CARD_NUM - Unable to find the specified controller ID
```

To view a list of commands at the command line, type `ARCCONF` and press `Enter`.

To access the online help for a specific command, type `ARCCONF <command>`, then press `Enter`.

## 2.1 ARCCONF Commands

The following commands are available in ARCCONF for Smart Storage controllers. The commands are described on the following pages, in alphabetical order. In the command descriptions, <> indicates a required parameter and [] indicates an optional parameter.

**Table 2 • ARCCONF Commands for Smart Storage Controllers**

atapassword	imageupdate	saveconfig	setstate
consistencycheck	key	savesupportarchive	setstatsdatacollection
create	list	setarrayparam	slotconfig <sup>2</sup>
delete	maxcrypto <sup>1</sup>	setboot	smp
driverupdate	maxcryptoaccounts <sup>1</sup>	setcache	splitmirror
expanderlist	maxcryptokey <sup>1</sup>	setconfig	task
expanderupgrade	modify	setconnectormode	uninit
getconfig	passthrough <sup>2</sup>	setcontrollermode	
getlogs	phyerrorlog	setcontrollerparam	
getsmartstats	playconfig	setmaxcache	
getstatus	rescan	setname	
getversion	resetstatisticscounters	setperform	
identify	romupdate	setpower	
		setpriority	

**Note:** ARCCONF supports commands for other controllers that are not listed in the table above. If you attempt to execute any command not listed [Table 2 • ARCCONF Commands for Smart Storage Controllers](#) on page 4, the firmware returns an error.

<sup>1</sup> Available on controllers that support maxCrypto Controller-Based Encryption. See the Release Notes for more information.

<sup>2</sup> Available in UEFI/ARCCONF only. See [Running ARCCONF in the UEFI Shell](#) on page 48.

## 2.2 arccnf atapassword

### Description

Sets or clears the password for SATA drives.

### Syntax

```
ARCCONF ATAPASSWORD <Controller#> SET <new password> <Channel# ID#> [nologs]
ARCCONF ATAPASSWORD <Controller#> CLEAR <current password> <Channel# ID#>
[nologs]
```

### Parameters

#### new password | current password

New password, current password.

#### Channel/ID

Lists the space-delimited channel number and device number (ID) pairs for each drive on which to set or clear the password.

### Examples

```
ARCCONF ATAPASSWORD 1 SET uR8ryx 0 1
ARCCONF ATAPASSWORD 1 CLEAR uR8ryx 0 1
```

## 2.3 arccnf consistencycheck

### Description

Toggles the background consistency check modes of the controller.

### Syntax

```
ARCCONF CONSISTENCYCHECK <Controller#> <on [Delay]|off> [noprompt] [nologs]
ARCCONF CONSISTENCYCHECK <Controller#> PARALLELCOUNT <Count> noprompt] [nologs]
ARCCONF CONSISTENCYCHECK <Controller#> EVENTNOTIFY <Enable | Disable> noprompt]
[nologs]
ARCCONF CONSISTENCYCHECK <Controller#> INCONSISTENCYREPAIRPOLICY <Enable |
Disable> noprompt] [nologs]
```

### Parameters

#### Controller#

Controller number.

#### On [Delay]

Turns background consistency check on, with optional 1 second–30 second delay period. The delay period sets the controller idle time, after which the consistency check will start. A value of 0 disables the consistency check (effectively the same as setting the parameter to Off). If Delay is unspecified, the consistency check mode is set to HIGH. If Delay is specified, the consistency check mode is set to IDLE for the specified period.

#### PARALLELCOUNT <Count>

Sets the parallel consistency check count. A value of 1 disables the consistency check.

**EVENTNOTIFY <Enable | Disable>**

Sets the inconsistency event notification and serial debug message generation setting for mirrored volumes. The Enable option enables the event notification and serial debug message generation. The Disable option disables the event notification and serial debug message generation.

**INCONSISTENCYREPAIRPOLICY <Enable | Disable >**

Sets the inconsistency repair policy for the controller. The Enable option enables the inconsistency repair; the Disable option disables the inconsistency repair.

**Noprompt**

Optional parameter that suppresses the confirmation prompt.

**Examples**

```
ARCCONF CONSISTENCYCHECK 1 OFF
ARCCONF CONSISTENCYCHECK 1 PARALLELCOUNT 4
ARCCONF CONSISTENCYCHECK 1 EVENTNOTIFY enable
ARCCONF CONSISTENCYCHECK 1 INCONSISTENCYREPAIRPOLICY enable
```

## 2.4 arconf create

**Description**

Creates a new encrypted or plaintext logical drive and, optionally, enables logical drive read caching, write caching. You must provide the channel and device ID of the physical devices.

On redundant logical drives, ARCCONF performs autosynchronization.

**Note:** Do not mix SMR and PMR drives in an array.

**Syntax**

```
ARCCONF CREATE <Controller#> LOGICALDRIVE [Options] <Size> <RAID#> <CHANNEL#
ID#> [CHANNEL# ID#] ... [noprompt] [nologs]
ARCCONF CREATE <Controller#> LOGICALDRIVE [Options] <Size> <RAID#> ARRAY
<Array#> [noprompt] [nologs]
ARCCONF CREATE <Controller#> LOGICALDRIVE ENCODE <Enable/Disable> USERROLE
<userrole> [PASSWORD <password>] [Options] <Size> <RAID#> ARRAY <Array#>
[noprompt] [nologs]
ARCCONF CREATE <Controller#> MAXCACHE [Options] DATALD, <LogicalDrive#> <Size>
<RAID#> <CHANNEL# ID#> [Channel# ID#]... [noprompt] [nologs]
ARCCONF CREATE <Controller#> MAXCACHE [Options] DATALD, <LogicalDrive#> <Size>
<RAID#> ARRAY <maxCache Array#> [noprompt] [nologs]
ARCCONF CREATE <Controller#> RAIDZEROARRAY <Channel# ID#> [Channel# ID#] ...
[noprompt] [nologs]
```

**Parameters****Controller#**

The controller number.

**Logical Drive, maxCache**

Indicates a logical drive or maxCache Device, with the following options:

- Stripesize <STRIPE>—Allows the logical drive stripe size to be built. Optional parameters for specifying a stripe size. STRIPE is specified in kilobytes 16, 32, 64, 128, 256, 512 and 1024 are supported. The default is 256 kB.
- Legs <LEG>—Optional parameters for specifying number of legs. Value is an integer.
  - LEG—Number of legs for RAID level 50 or 60.

- Default—2 legs
  - RAID 50—2-16 legs, 3-32 drives/leg, 128 drives max.
  - RAID 60—2-16 legs, 4-16 drives/leg, 128 drives max.
- Name <NAME>—Optional parameter for specifying the alias name of a logical device that is displayed in the utilities. Value is a string of up to 64 characters.
  - Method <METHOD>—Initialization method for the logical drive. Valid options include: BUILD DEFAULT, DEFAULT.
  - LDcache—Sets the cache state for the logical drive:
    - LON— cache on
    - LOFF—cache off
  - SSD I/O BYPASS <enable | disable>—Specifies to enable/disable SSD I/O bypass path on an array.
  - CacheLineSize <CACHELINESIZE>—Specifies the cache line size in KB for maxCache. Adjusting the cache line size can impact maxCache performance and maximum size supported. The larger cache line size can support the larger maxCache size. The default value for this parameter is 64.
  - Wcache—Sets the logical drive write cache mode for maxCache devices:
    - WT— write-through disabled
    - WB—write-back enabled
    - WBB—write-back enabled (when protected by battery or flash backup module)

### Data Logical Drive #

Specifies the existing data logical drive number to associate with the newly created cache logical device.

### Encode <enable/disable>

Creates encrypted or plaintext logical drives, based on the maxCrypto™ status and Mixed Volumes logical device properties (see notes below; see also [arcconf maxcrypto](#) on page 17):

- Enable—Creates an encrypted logical drive.
- Disable—Creates a plaintext logical drive.

#### Note:

1. If maxCrypto status is Disabled, then only plaintext logical drives can be created.
2. If maxCrypto status is Enabled and Mixed Volumes property is Enabled, both encrypted and plaintext logical drives can be created.
3. If maxCrypto status is Enabled and Mixed Volumes property is Disabled, only encrypted logical drives can be created.
4. If maxCrypto status is Enabled, then logical drives are encrypted by default.
5. To create plaintext logical drives, the Encode option must be specified with authentication credentials (Userrole/Password).

### Userrole <userrole> [Password <password>]

maxCrypto user-role and password. Valid values are:

- crypto (maxCrypto administrator)
- user (standard user)

### Array <Array#>

Array number on which to create the logical drive.

### ARRAY <maxCache Array#>

The maxCache array number.

### RAIDZEROARRAY

Create arrays from list of physical device(s) specified.

Each array will contain exactly one physical device and one RAID 0 logical device.

Default values will be applied for all logical device(s) created.

### SSDOverProvisioningOptimization <enable | disable>

Specifies to initialize solid state drives that support the rapid parity initialization feature.

#### Size

Indicates the size of the logical drive in megabytes. Use MAX to set size to available space. Use MAXMBR to set the size to 2 TB.

#### RAID#

Indicates the RAID level for the new logical drive: 0, 1, 10, 1(ADM), 10(ADM), 50, 60, and 6(ADG) are supported.

**Note:** For a complete list of supported RAID levels for your controller, refer to the product release notes.

#### Channel# ID#

Lists the space-delimited channel number and device number pairs for each device to add to the logical drive.

#### Noprompt

No prompt for confirmation.

#### Examples

```
ARCCONF CREATE 1 LOGICALDRIVE STRIPESIZE 64 MAX 0 1 0 2 0 3 2 NOPROMPT
ARCCONF CREATE 1 LOGICALDRIVE ssdoverprovisioningoptimization enable 1024 0 ARRAY 0
```

```
ARCCONF CREATE 1 LOGICALDRIVE 1024 1 ARRAY 0
ARCCONF CREATE 1 LOGICALDRIVE stripesize 16 method build MAX 5 0 0 0 1 0 2
ARCCONF CREATE 1 MAXCACHE WB dataid 0 17000 1 0 0 0 1
ARCCONF CREATE 1 MAXCACHE dataid 0 17000 0 ARRAY 0
ARCCONF CREATE 1 LOGICALDRIVE ENCODE disable USEROLE crypto PASSWORD Abc@1234 MAX 5 0
0 0 1 0 2
```

## 2.5 arccnf delete

### Description

Deletes a logical drive, an array, or maxCache logical device . All data stored on the logical drive will be lost.

### Syntax

```
ARCCONF DELETE <Controller#> LOGICALDRIVE <LD#> <LD#> ...|ALL [noprompt]
[nologs]
ARCCONF DELETE <Controller#> LOGICALDRIVE ALL [noprompt] [nologs]
ARCCONF DELETE <Controller#> ARRAY <arr#> [noprompt] [nologs]
ARCCONF DELETE <Controller#> ARRAY ALL [noprompt] [nologs]
ARCCONF DELETE <Controller#> MAXCACHE <maxCache ld#> [noprompt] [nologs]
ARCCONF DELETE <Controller#> MAXCACHE ALL [noprompt] [nologs]
```

### Parameters

#### Controller#

Controller# is the controller number.

#### LD#

LogicalDrive# is the number of the logical drive to be deleted.

#### arr#

arr# is the number of the array to be deleted.

#### maxCache ld#

maxCache ld# is the number of the maxCache logical device to be deleted.

#### ALL

Deletes all logical devices on array or maxCache.



**Noprompt**

Optional parameter that suppresses alert messages.

**Examples**

```
ARCCONF DELETE 1 LOGICALDRIVE 1 2 3
ARCCONF DELETE 1 ARRAY 0
ARCCONF DELETE 1 ARRAY ALL
```

## 2.6 arconf driverupdate

**Description**

Updates the Windows device driver for the controller.

**Note:** This command is available on Windows systems only.

**Syntax**

```
ARCCONF DRIVERUPDATE <DirName> [nologs]
```

**Parameters****DirName**

Absolute path to directory containing the Windows driver for the controller.

**Nologs**

Optional parameter that suppresses log output.

**Examples**

```
ARCCONF DRIVERUPDATE C:\WINDOWSALL
```

## 2.7 arconf expanderlist

**Description**

Returns a list of disk drive expanders on a controller.

**Syntax**

```
ARCCONF EXPANDERLIST <Controller#> [nologs]
```

**Parameters****Controller#**

Controller number.

## Examples

```
ARCCONF EXPANDERLIST 1
```

## 2.8 arconf expanderupgrade

### Description

Allows new firmware to be flashed to an enclosure or expander.

### Syntax:

```
ARCCONF EXPANDERUPGRADE <Controller#> ENCLOSURE <Connector# Channel# ID#>
[ChunkSize#] <UpgradeType> <Filename> [Mode#] [noprompt] [nologs]
```

### Parameters

#### Controller#

Controller number.

#### Channel#

Channel number of the device to be updated.

#### ID#

Device number of the device to be updated.

#### Connector#

Connector number of the device to be updated.

#### ChunkSize#

Chunk size, in bytes, to be used to update the firmware. Default is 4096 bytes.

#### Filename

Name of the firmware update file.

#### UpgradeType

EXPANDER—update the firmware image on the expander or enclosure.

MFG—update the manufacturing image (BOOT SEEPROM) on the expander or enclosure.

CPLD—update the CPLD image on the expander or enclosure.

**Note:** MFG and CPLD upgrade types are supported on the Microsemi Adaptec AEC-82885T expander only.

#### Mode#

The Mode parameter applies to EXPANDER and MFG upgrade types only. Valid values are:

- 2—download microcode only; requires system reset or power cycle to activate (default).
- 6—download microcode with offsets and activate.
- 7—download microcode with offsets, save, and activate.
- E—download microcode with offsets and defer activation.
- F—activate deferred microcode. It does not require the filename as an input.

#### Noprompt

Optional parameter that suppresses alert messages.

## Examples

```
arcconf EXPANDERUPGRADE 1 ENCLOSURE 2 0 0 1024 EXPANDER C:\FirmwareImage.bin 7
arcconf EXPANDERUPGRADE 1 ENCLOSURE 2 0 0 512 MFG C:\FirmwareImage.rom 6
arcconf EXPANDERUPGRADE 1 ENCLOSURE 2 0 0 256 CPLD C:\CPLDImage.bin noprompt
```

## 2.9 arconf getconfig

### Description

Lists the following information:

- Array status, size and member drives
- Controller type, status, World Wide Name (WWN), manufacturing information, and mode
- Cache preservation status: enabled/disabled, % of cache pages preserved
- BIOS, boot block, device driver, and firmware versions
- Logical drive status, RAID level and size
- Logical drive mount points
- RAID 10, 50, 60 segment and group information
- maxCache status, SSD information, and statistics of the maxCache logical drive
- Device type, device ID, presence of PFA
- Physical device state, mount point (for drives with OS partition)
- Enclosure information: fan, power supply, and temperature status
- SGPIO virtual SEP information (virtual enclosure device for SGPIO backplanes)
- Connectors backplane discovery protocol
- Connector/Lane/Phy mapping
- Green backup details
- I2C address, clock speed, and clock stretching status
- maxCrypto properties: status, mode, number of encrypted logical devices, master key configuration, account configuration
- Out-of-Band interface properties

Also displays controller BIOS settings if you do not include a device-type keyword.

**Note:** When displaying adapter information (AD keyword), the Controller Status field is set to `Ok` or `Not Ok`. Its value is set to `Not Ok` only if:

1. Communication with the controller fails. This occurs when the driver returns an error code after attempting to send a command to the controller.
2. A logical drive was created with a newer version of `arcconf`. Update to the latest utilities.
3. The controller mode (RAID/Mixed/HBA) is supported by the hardware, but not the firmware. Usually, this means that an older version of `arcconf` is being used against a newer controller. Update to the latest utilities.

### Syntax

```
ARCCONF GETCONFIG <Controller#> [AD|LD [LD#]] AR[AR#]|PD [Channel# ID# Channel#
ID#...]|MC|CN| [AL]] [nologs]
```

### Parameters

#### Controller#

Controller number

### LD#

Display information about the specified logical device

### AR#

Display information about the specified array, including the associated split mirror array, if applicable

### AD/PD/AL...

- AD—Adapter information only (including maxCrypto properties)
- LD—Logical drive information only
- AR—Array information only
- PD—Physical device information only
- MC—maxCache information only
- CN—Connector information only
- AL—All information
- LD#—Optionally displays information about the specified logical device
- AR#—Optionally displays information about the specified array

### Channel# ID#

Channel# ID#: The Channel and ID of the physical device to be display.

### Examples

```
arcconf getconfig 1
Controllers found: 1

-----
Controller information
-----
Controller Status           : Optimal
Controller Mode             : Mixed
Channel description        : SCSI
Controller Model           : MSCC Adaptec SmartRAID 3162-8i /e
Controller Serial Number   : 7139F300097
Controller World Wide Name : 50000D1E001C7D80
Physical Slot               : 6
Temperature                : 62 C/ 143 F (Normal)
Host bus type              : PCIe 3.0
Host bus speed              : 7880 MBps
Host bus link width        : 8 bit(s)/link(s)
PCI Address (Bus:Device:Function) : 1:0:0
Number of Ports            : 2
Internal Port Count        : 2
External Port Count        : 0
Defunct disk drive count   : 0
NCQ status                  : Enabled
Queue Depth                : Automatic
Monitor and Performance Delay : 60 minutes
Elevator Sort              : Enabled
Degraded Mode Performance Optimization : Disabled
Latency                    : Disabled
Statistics data collection mode : Disabled
Post Prompt Timeout        : 15 seconds
Boot Controller            : False
Primary Boot Volume        : None
Secondary Boot Volume      : None
Driver Name                : SmartPqi.sys
Driver Supports SSD I/O Bypass : Yes
Manufacturing Part Number  : Not Applicable
Manufacturing Spare Part Number : Not Applicable
Manufacturing Wellness Log  : Not Applicable
NVRAM Checksum Status      : Passed
Sanitize Lock Setting      : Anti-Freeze

-----
Power Settings
-----
Power Consumption          : Not Available
Current Power Mode         : Maximum Performance
Pending Power Mode         : Not Applicable
Survival Mode              : Enabled
```

## 2.10 arccconf getlogs

### Description

Provides access to event logs including:

- A log of special events that may have occurred (rebuilds, LDMs, and so on)

### Syntax

```
ARCCONF GETLOGS <Controller#> <Type1> [tabular] [nologs]
ARCCONF GETLOGS <Controller#> <Type2> [tabular] [nologs]
ARCCONF GETLOGS <Controller#> <Type2> LOGICALDRIVE <LD#> [tabular] [nologs]
ARCCONF GETLOGS <Controller#> DEVICE <clear> <Channel# ID#> [nologs]
ARCCONF GETLOGS <Controller#> DEVICE <clear> ALL [nologs]
```

### Parameters

#### Controller#

Controller number.

#### Type1

- EVENT—controller event log

#### Type2

- STATS—controller statistics data

#### Type3

- CACHE—cache statistics data for all or a single logical drive

#### Clear

Clears the error counter for one or all physical drives on a controller.

Clear device error counter is not applicable on 'Not Supported' drives.

#### Channel/ID

Channel and number of the physical device on the controller.

#### Tabular

Displays statistics in tabular (vs XML) format.

### Examples

```
ARCCONF GETLOGS 1 EVENT tabular
ARCCONF GETLOGS 1 STATS tabular
ARCCONF GETLOGS 1 STATS LOGICALDRIVE 0 tabular
```

## 2.11 arccconf getsmartstats

### Description

Displays SMART statistics for the hard drives and Solid State Drives (SSDs) on a controller.

### Syntax

```
ARCCONF GETSMARTSTATS <Controller#> [Tabular] [nologs]
```

**Parameters****Controller#**

Controller number.

**Tabular**

Creates output in tabular format.

**Examples**

```
ARCCONF GETSMARTSTATS 1
ARCCONF GETSMARTSTATS 1 TABULAR
```

## 2.12 **arccconf getstatus**

**Description**

The GETSTATUS function displays the status of any background task that is currently running. The information includes the type of operation, status, logical drive number and logical drive size (for a logical device), channel ID/device ID (for a physical drive), and percentage of the operation completed.

**Syntax**

```
ARCCONF GETSTATUS <Controller#> [nologs]
```

**Parameters****Controller#**

Controller# is the controller number

**Examples**

```
ARCCONF GETSTATUS 1
```

## 2.13 **arccconf getversion**

**Description**

Lists version information for all controllers or a specific controller's software components, including information about the driver and firmware currently running.

**Syntax**

```
ARCCONF GETVERSION [nologs]
ARCCONF GETVERSION <Controller#> [nologs]
```

**Parameters****Controller#**

Controller# is the controller number

## Examples

```
ARCCONF GETVERSION
ARCCONF GETVERSION 1
```

## 2.14 arccnf identify

### Description

Identifies a physical device by blinking its LEDs. Also, identifies all the physical devices that are used to create a logical drive, maxCache or array.

### Syntax

```
ARCCONF IDENTIFY <Controller#> ALL [TIME <BlinkTime>] [STOP] [nologs]
ARCCONF IDENTIFY <Controller#> LOGICALDRIVE <LogicalDrive#> [TIME <BlinkTime>]
[nologs]
ARCCONF IDENTIFY <Controller#> DEVICE <Channel# ID#> ... [TIME <BlinkTime>]
[nologs]
ARCCONF IDENTIFY <Controller#> ARRAY <Array#> [TIME <BlinkTime>] [nologs]
ARCCONF IDENTIFY <Controller#> MAXCACHE [TIME <BlinkTime>] [nologs]
```

### Parameters

#### Controller#

Controller number

#### LogicalDrive#

Number of the logical drive to be identified

#### Array#

Array number

#### MAXCACHE

maxCache device to be identified

#### Channel# ID#

Channel number and ID number for the physical device(s) to be identified

#### ALL

Blinks all physical devices on the controller for 1 hour or until the STOP command is issued

#### TIME <BlinkTime>

Time, in seconds, for the LEDs to continue blinking

#### STOP

Stops blinking the device

### Examples

```
ARCCONF IDENTIFY 1 DEVICE 0 0
ARCCONF IDENTIFY 1 ALL TIME 60
ARCCONF IDENTIFY 1 ALL STOP
ARCCONF IDENTIFY 1 LOGICALDRIVE 0 TIME 60
ARCCONF IDENTIFY 1 LOGICALDRIVE 0
ARCCONF IDENTIFY 1 DEVICE 0 1 TIME 30
ARCCONF IDENTIFY 1 ARRAY 0
```

## 2.15 **arcconf imageupdate**

### Description

Allows new firmware to be flashed to the hard drive.

### Syntax:

```
ARCCONF IMAGEUPDATE <Controller#> DEVICE <Channel# ID# ChunkSize# Filename>
[Mode#] [BufferID#] [noprompt] [nologs]
```

### Parameters

#### Controller#

Controller number.

#### Channel#

Channel number of the device to be updated.

#### ID#

Device number of the device to be updated.

#### ChunkSize#

Chunk size, in bytes, to be used to update the firmware.

**Note:** For SATA drives, the chunk size must be a multiple of 512.

#### Filename

Name of the firmware update file.

#### Mode#

Firmware update mode. Valid values for physical drives are:

- 3—Download with offsets and save image for immediate and future use
- 7—Download microcode with offsets, save, and activate
- 5—Download microcode in single transfer and activate
- 14(E)—Download microcode in 'ChunkSize' byte chunks, but do not activate
- 239(E+F)—Download microcode in 'ChunkSize' byte chunks and activate

#### BufferID#

Mandatory for tape drive firmware update.

#### Noprompt

Optional parameter that suppresses alert messages.

### Examples

```
ARCCONF IMAGEUPDATE 1 DEVICE 0 0 32768 ados.lod 3
```

## 2.16 **arcconf key**

### Description

Loads a feature key onto a controller.



**Syntax**

```
ARCCONF KEY <Controller#> SET <Key#> [nologs]
```

**Parameters****Controller#**

The controller number.

**Key#**

The key number provided by Microsemi.

**Examples**

```
ARCCONF KEY 1 SET ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

## 2.17 arconf list

**Description**

Lists all controllers in the system, or the configuration of a specific controller.

**Syntax**

```
ARCCONF LIST [Controller#] [nologs]
```

**Parameters****Controller#**

The controller number.

**Examples**

```
ARCCONF LIST  
ARCCONF LIST 1
```

## 2.18 arconf maxcrypto

**Description**

Configures maxCrypto settings, including:

- maxCrypto master key
- Mode (enable/disable)
- Administrator account credentials
- Support for mixed encrypted/plain text volumes

Also toggles the maxCrypto mode, encodes/encrypts arrays and logical drives, and shows the maxCrypto certificate.

## Syntax

```

ARCCONF MAXCRYPTO <Controller#> SETUP manual MODE <enable <ACCEPT <yes | no>
> | disable> KEYMANAGEMENTMODE local MIXEDVOLUMES <enable | disable> MASTERKEY
<masterkeystring> USERROLE crypto [PASSWORD <crypto password>] [nologs]

ARCCONF MAXCRYPTO <Controller#> MODE <enable <ACCEPT <yes | no> > | disable>
USERROLE <crypto | user> [PASSWORD <crypto/user password>] [nologs]
ARCCONF MAXCRYPTO <Controller#> ENCODE LOGICALDRIVE <logicaldrive#> DATA
<preserve/discard> USERROLE <crypto | user> [PASSWORD <crypto/user password>]
[nologs]
ARCCONF MAXCRYPTO <Controller#> ENCODE ARRAY <array#> DATA <preserve/discard>
USERROLE <crypto | user> [PASSWORD <crypto/user password>] [nologs]
ARCCONF MAXCRYPTO <Controller#> SHOW certificate [nologs]

```

## Parameters

### Controller#

Controller number.

### SETUP manual

Enables manual setup; all parameters are required.

### MODE <enable <ACCEPT <yes | no> > | disable>

Enables creation of encrypted and plaintext logical devices and allows you to accept the maxCrypto Terms of Use. Valid values are:

- Enable: Authorized users can create encrypted logical devices or plaintext logical devices, based on the value of the MIXEDVOLUMES property.
- Disable: Authorized users can create plaintext (non-encrypted) logical devices only.

### KEYMANAGEMENTMODE local

Enables local key management.

### MIXEDVOLUMES

Enables mixing of encrypted and plaintext logical devices. Valid values are:

- Enable: Authorized users have the option to create encrypted logical devices or plaintext logical devices.
- Disable: New logical devices will be encrypted, with no option to create plaintext logical devices.

### MASTERKEY <masterkeystring>

A 10 to 32 character string, using all printable ASCII characters.

**Important:** Be sure to record the master key and store in a safe place. Once set, the master key cannot be displayed or recovered, only reset.

### USERROLE <userrole> [PASSWORD <password>]

maxCrypto user role and password. Valid values are:

- crypto (maxCrypto administrator)
- user (standard user)

The password is a 8-16 character string, comprising all printable ASCII characters. It must include at least one uppercase character, one lowercase character, one numeric, and one special character (#,!,@,.,...). If password is not entered on the command line, a prompt appears during command execution.

### ENCODE

Encrypts an existing array or logical drive, based on the maxCrypto mode and MIXEDVOLUMES property.

### DATA <preserve | discard>

Preserves or discards original data in encoded logical device.

### SHOW certificate

Displays the maxCrypto Terms of Use certificate.

## Examples

```
ARCCONF MAXCRYPTO 1 SETUP manual MODE enable ACCEPT yes KEYMANAGEMENTMODE local
MIXEDVOLUMES enable MASTERKEY xxxxxx USERROLE crypto PASSWORD xxxxxx
ARCCONF MAXCRYPTO 1 SETUP manual MODE disable KEYMANAGEMENTMODE local MIXEDVOLUMES enable
MASTERKEY xxxxxx USERROLE crypto password xxxxxx
ARCCONF MAXCRYPTO 1 MODE disable USERROLE crypto PASSWORD xxxxxx
```

## 2.19 arconf maxcryptoaccounts

### Description

Creates a maxCrypto standard user (non-administrator) account, sets password recovery question/answer, changes passwords, and recovers passwords.

### Syntax

```
ARCCONF MAXCRYPTOACCOUNTS <Controller#> CREATEUSER [CRYPTOPASSWORD <crypto
password> USERPASSWORD <user password>][nologs]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> CHANGEPASSWORD USERROLE <crypto | user>
[OLDPASSWORD <crypto/user password> NEWPASSWORD <crypto/user password>]
[nologs]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> SETRECOVERYPARAM QUESTION <"Question">
ANSWER <"Answer"> USERROLE crypto [PASSWORD <crypto password>][nologs]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> RECOVERPASSWORD ANSWER <"Answer">
USERROLE crypto [NEWPASSWORD <crypto password>][nologs]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> RECOVERPASSWORD SHOW question [nologs]
```

### Parameters

#### Controller#

Controller number.

#### CREATEUSER

Creates a standard user account, using the maxCrypto Administrator account (crypto).

**Note:** The standard user account is limited to lock/unlock firmware update; see [arconf setcontrollerparam](#) on page 34.

#### CRYPTOPASSWORD <crypto password>

maxCrypto Administrator account (crypto) password. If crypto password is not entered on the command line, a prompt appears during command execution.

#### USERPASSWORD <user password>

maxCrypto standard account (user) password. The password is a 8-16 character string, comprising all printable ASCII characters. It must include at least one uppercase character, one lowercase character, one numeric, and one special character (#,!,@,...). If user password is not entered on the command line, a prompt appears during command execution.

#### CHANGEPASSWORD

Changes the password for the standard user or crypto (Administrator) account.

#### USERROLE <crypto | user>

The account type: crypto (Administrator) or user (standard user).

#### OLDPASSWORD <crypto/user password> NEWPASSWORD <crypto/user password>

The old password and new password for the crypto account or user account. The password is a 8-16 character string, comprising all printable ASCII characters. It must include at least one uppercase character, one lowercase character, one numeric, and one special character (#,!,@,...). If the password is not entered on the command line, a prompt appears during command execution.

#### **SETRECOVERYPARAM QUESTION <"Question"> ANSWER <"Answer">**

Sets the password recovery question and answer for the crypto (Administrator) account. The question and answer must be enclosed in quotes.

#### **RECOVERPASSWORD ANSWER <"Answer"> [NEWPASSWORD <crypto password>]**

Answers the recovery question and sets the new password for the crypto (Administrator) account. If the password is not entered on the command line, a prompt appears during command execution.

#### **RECOVERPASSWORD SHOW question**

Shows the recovery question.

#### **Examples**

```
ARCCONF MAXCRYPTOACCOUNTS 1 CHANGEPASSWORD USERROLE crypto OLDPASSWORD Abc@1234
NEWPASSWORD Abc@123456
ARCCONF MAXCRYPTOACCOUNTS 1 CREATEUSER CRYPTOPASSWORD Abc@1234 USERPASSWORD Abc@123456
ARCCONF MAXCRYPTOACCOUNTS 1 SETRECOVERYPARAM QUESTION "Which planet are you from?" ANSWER
  "i am from planet earth" USERROLE crypto PASSWORD Abc123456
```

## **2.20 arccnf maxcryptokey**

### **Description**

Performs maxCrypto key management functions, including changing the master key, generating a new key for an encrypted array or logical drive, and importing a master key for a logical drive moved from another controller (allows the controller to access the encrypted data).

### **Syntax**

```
ARCCONF MAXCRYPTOKEY <Controller#> CHANGEMASTERKEY <masterkey> USERROLE <crypto
| user> [PASSWORD <crypto/user password>][nologs]
ARCCONF MAXCRYPTOKEY <Controller#> REKEY ARRAY <array#> USERROLE <crypto |
user> [PASSWORD <crypto/user password>][nologs]
ARCCONF MAXCRYPTOKEY <Controller#> REKEY LOGICALDRIVE <logicaldrive# | ALL>
USERROLE <crypto | user> [PASSWORD <crypto/user password>][nologs]
ARCCONF MAXCRYPTOKEY <Controller#> IMPORT MASTERKEY <masterkey> USERROLE <crypto
| user> [PASSWORD <crypto/user password>][nologs]
```

### **Parameters**

#### **Controller#**

Controller number.

#### **CHANGEMASTERKEY <masterkeystring>**

A 10 to 32 character string, using all printable ASCII characters.

**Important:** Be sure to record the new master key and store in a safe place. Once set, the master key cannot be displayed or recovered, only reset.

#### **IMPORT MASTERKEY <masterkeystring>**

Imports the master key for a logical drive moved from another controller. The master key is a 10 to 32 character string, using all printable ASCII characters.

## REKEY

Generates a new key for an encrypted array or logical drive.

### USEROLE <crypto | user> [PASSWORD <crypto/user password>]

maxCrypto user role: crypto (Administrator) or user (standard account), with optional password. If password is not entered on the command line, a prompt appears during command execution.

### Examples

```
ARCCONF MAXCRYPTOKEY 1 CHANGEMASTERKEY Abc@1234567 USEROLE crypto PASSWORD Abc@123456
ARCCONF MAXCRYPTOKEY 1 REKEY ARRAY 0 USEROLE crypto PASSWORD Abc@123456
```

## 2.21 arconf modify

### Description

Morphs a logical device from one RAID level to another (RAID Level Migration). Expands a logical device from original size to one with larger capacity (Online Capacity Expansion).

Expands, shrinks or moves an array, or moves a logical device to a new array.

**Note:** Do not mix SMR and PMR in an array.

### Syntax

```
ARCCONF MODIFY <Controller#> FROM <LogicalDrive#>
TO [Options] <Size> <RAID#> <CHANNEL# ID#> [CHANNEL# ID#] ... [noprompt]
[nologs]
```

```
ARCCONF MODIFY <Controller#> ARRAY <Array#> MOVE <Channel# ID#> [Channel# ID#]
... [nologs]
ARCCONF MODIFY <Controller#> ARRAY <Array#> HEAL <Channel# ID#> [Channel# ID#]
... [nologs]
ARCCONF MODIFY <Controller#> LOGICALDRIVE <LD#> MOVEARRAY <Array#> [nologs]
ARCCONF MODIFY <Controller#> LOGICALDRIVE <LD#> NEWARRAY <Channel# ID#>
[Channel# ID#] ... [nologs]
ARCCONF MODIFY <Controller#> ARRAY <Array#> EXPAND [modifyparitygroups]
<Channel# ID#> [Channel# ID#] ... [nologs]
ARCCONF MODIFY <Controller#> ARRAY <Array#> SHRINK [modifyparitygroups]
<Channel# ID#> [Channel# ID#] ... [nologs]
```

### Parameters

#### Controller#

The controller number

#### LogicalDrive#

The logical drive number to be modified

#### Array#

The array ID of the array to be modified

#### Options

One of the following:

- Stripesize <size>—indicates the stripe size in KB. Options are 16, 32, 64, 128, 256, 512, and 1024. the default is 256KB.

#### Size

- Size in MB.

- MAX indicates that you want to use all available space on the disk.

**RAID#**

RAID level for the logical drive: 0, 1, 10, 50 and 60 are supported.

**Channel# ID#**

Channel number and device ID for the device

**Note:** The CHANNEL# and ID# parameters are the list of devices that will contain the target modification object. Channel and ID are repeatable parameters. For RAID 1 to Simple Volume migration, CHANNEL# and ID# parameters are ignored.

**MOVE**

Moves an array to a new set of physical devices. Number of new physical devices must equal the number of physical devices in the original array.

**EXPAND**

Expands an array by adding physical device(s) to it. Only the physical device(s) that need to be added should be specified.

**SHRINK**

Shrinks an array by removing physical device(s) from it. Only physical device(s) that need to be removed should be specified.

**HEAL**

Replaces failed physical devices in the array with the specified devices

**MOVEARRAY**

Moves a logical device to an existing array

**NEWARRAY**

Moves a logical device to a new array created with the specified physical devices

**modifyparitygroups**

Reconfigures the logical device(s) parity groups based on the final number of physical devices in the array

**noprompt**

Suppresses the user prompt

**Examples**

```
ARCCONF MODIFY 1 FROM 2 TO 2048 0 0 123 0 124 0 117
ARCCONF MODIFY 1 ARRAY 1 MOVE 0 2 0 3
ARCCONF MODIFY 1 ARRAY 1 HEAL 0 0 0 1
ARCCONF MODIFY 1 LOGICALDRIVE 0 MOVEARRAY 1
ARCCONF MODIFY 1 LOGICALDRIVE 0 NEWARRAY 0 4 0 5
ARCCONF MODIFY 1 ARRAY 1 EXPAND 0 0 0 1
ARCCONF MODIFY 1 ARRAY 1 SHRINK 0 0 0 1
ARCCONF MODIFY 1 ARRAY 1 EXPAND MODIFYPARITYGROUPS 0 0 0 1
ARCCONF MODIFY 1 ARRAY 1 SHRINK MODIFYPARITYGROUPS 0 0 0 1
```

## 2.22 arconf passthrough

**Description**

Sends a passthrough SCSI command. The CDB bytes are enclosed in square brackets, hex encoded, space-delimited, and must number 6, 10, 12, or 16 bytes. Read data may be redirected to a file. Write data is taken from the specified file. Transfers are limited to 2048 bytes. Transfer lengths are inferred for common SCSI CDBs but may be overridden using the length parameter.

## Syntax

```

ARCCONF PASSTHROUGH <Controller#> <Channel# ID#> <read|nottransfer> [length]
<[> CDB> <]> [rawhex] [noprompt] [nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#>
<readwithsense|nottransferwithsense> [length] <[> <CDB> <]> [rawhex] [noprompt]
[nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#> <write> [length] <[> <CDB>
<]> <Filename> [rawhex] [noprompt] [nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#> <writewithsense> [length] <[>
<CDB> <]> <Filename> [rawhex] [noprompt] [nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#1-ID#2> <read> [length] <[>
<CDB> <]> [rawhex] [noprompt] [nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#1, ID#2, #ID#3> <read> [length]
<[> <CDB> <]> [rawhex] [noprompt] [nologs]

```

## Parameters

### CDB

SCSI Command Descriptor Block. The CDB bytes are enclosed in square brackets, hex encoded, space-delimited, and must be 6, 10, 12, or 16 bytes.

### length

Read/write data buffer length.

### Read

Command direction is read.

### Write

Command direction is write.

### nottransfer

No command direction (No data to read/write from/to the device).

### readwithsense

Command direction is read with sense data.

### writewithsense

Command direction is write with sense data.

### nottransferwithsense

Read only the sense data.

### rawhex

Displays Hex data only of the Passthrough response.

### Controller#

The controller through which the passthrough CDB is to be sent.

### Channel# ID#

The channel and ID of the physical device.

### FileName

Write the CDB data input file.

### noprompt

Suppress alert messages.

### nologs

Suppress log output.

### Supported Commands

Refer to the SCSI primary/block command specification for command format for each of the following commands.

**Table 3 • Passthrough CDB Commands**

Opcode	Command	Direction
0x00	TEST UNIT READY	nottransfer
0x03	REQUEST SENSE	read
0x08	READ (6)	read
0x0A	WRITE (6)	write
0x12	INQUIRY	read
0x15	MODE SELECT (6)	write
0x1A	MODE SENSE (6)	read
0x1B	START STOP UNIT	nottransfer
0x1C	RECEIVE DIAGNOSTIC RESULTS	read
0x1D	SEND DIAGNOSTIC	write
0x25	READ CAPACITY (10)	read
0x28	READ (10)	read
0x2A	WRITE (10)	write
0x2E	WRITE AND VERIFY (10)	write
0x3B	WRITE BUFFER	write
0x3C	READ BUFFER	read
0x4C	LOG SELECT	write
0x4D	LOG SENSE	read
0x55	MODE SELECT (10)	write
0x5A	MODE SENSE (10)	read
0x5E	PERSISTENT RESERVE IN	read
0x5F	PERSISTENT RESERVE OUT	write
0x85	ATA PASSTHROUGH (16)	write
0x88	READ(16)	read
0xA0	REPORT LUNS	read
0xA8	READ (12)	read
0xAA	WRITE (12)	write
0xAE	WRITE AND VERIFY (12)	write



## 2.23 arconf phyerrorlog

### Description

Displays PHY error logs for physical devices on a controller.

### Syntax

```
ARCCONF PHYERRORLOG <Controller#> DEVICE <Channel# ID#> [nologs]
ARCCONF PHYERRORLOG <Controller#> DEVICE ALL [nologs]
```

### Parameters

#### Controller#

Controller number.

#### Channel/ID

Channel and number of the physical device on the controller.

#### ALL

Displays PHY error log for all physical devices.

### Examples

```
ARCCONF PHYERRORLOG 1 DEVICE 0 0
ARCCONF PHYERRORLOG 1 DEVICE ALL
```

## 2.24 arconf playconfig

### Description

Configures a controller using a XML server template file produced by the SAVECONFIG command (see [arconf saveconfig](#) on page 28). Use this command to deploy the same controller configuration on multiple servers in your storage space.

#### Note:

1. The XML server template file (default, saveconfig.xml) is editable. For example, you may need to change the disk drive capacity, logical drive size, or RAID level.
2. Drives from the same vendor with slightly different capacities (147GB vs 150GB, for instance) are considered interchangeable. If the interchange results in a change in logical drive capacity, the drive is scaled, as needed. For example, if the new drives have 4% more capacity due to vendor or model changes, then all logical drives are increased in size by 4%.
3. Be sure to check the log file to verify that the controller was configured successfully. The exit codes, shown below, indicate the success or failure of the operation and if the system needs to be rebooted.

Code	Value	Meaning
SUCCESS	0	Configuration succeeded, no reboot is required.
FAILURE_GENERAL	1	An error occurred and the configuration could not be completed.

Code	Value	Meaning
SUCCESS_REBOOT	2	Configuration succeeded, but a reboot is required.

## Syntax

```
ARCCONF PLAYCONFIG <Input XML File> [LogFile] [FORCE ALL|LOGICALSIZE] [SLOTID][nologs]
```

## Parameters

### Input XML File

The pathname of the server template file. The default server template file is available at C:\PMCS\Log\saveconfig.xml.

### LogFile

Sets the pathname of the error log file. By default, the error log is available at C:\PMCS\Log\playconfig.log.

### FORCE

Forces deployment of the server even if the controller does not support all features, or the drive capacity does not match the configuration in the input XML file. Use FORCE ALL to force deployment of all features; use FORCE LOGICALSIZE to force deployment of just the logical drives.

### SLOTID

Apply the configuration based on Slot ID instead of Device ID.

## Examples

```
ARCCONF PLAYCONFIG server1_config.xml playconfig.log FORCE ALL
```

## 2.25 arconf rescan

### Description

Enables the controller to check for the removal of any disk drives and to check for the connection of any new disk drives to the controller. Controller rescan runs in the background, asynchronously. When rescan is started, a message is displayed stating that the process is running in the background and may take 10 minutes to complete. Another message is displayed if a rescan is started while one is already in progress.

### Syntax

```
ARCCONF RESCAN <Controller#> [nologs]
ARCCONF RESCAN ALL [nologs]
```

### Parameters

#### Controller#

The controller number

#### ALL

Rescans all controllers in the system

## Examples

```
ARCCONF RESCAN 1
ARCCONF RESCAN ALL
```

## 2.26 arconf resetstatisticscounters

### Description

Resets statistics counters for a controller and the logical and physical devices attached to it. Use this command to clear the counters and create fresh statistics, including (but not limited to):

- Read/Write Request Count
- Sectors Read/Written/Flushed
- Unaligned Reads/Writes
- Avg/Max Request Latency
- Max Queue Depth
- Max Request Latency
- Avg Dirty Cache Lines
- Avg Free Processor Ram
- Avg Locked Stripes
- Command Count

### Syntax

```
ARCCONF RESETSTATISTICSCOUNTERS <Controller#> [nologs]
```

### Parameters

#### Controller#

The controller number

### Examples

```
ARCCONF RESETSTATISTICSCOUNTERS 1
```

## 2.27 arconf romupdate

### Description

Allows new firmware and BIOS to be flashed to the controller. A reboot is required for the new firmware to take effect.

#### Note:

1. This command is supported on all OSs that support maxView Storage Manager.

## Syntax

```
ARCCONF ROMUPDATE <Controller#> <ImagePath> [once] [noprompt] [nologs]
ARCCONF ROMUPDATE 1 toggle [noprompt] [nologs]
```

## Parameters

### Controller#

The controller number.

### ImagePath

This is the full path of the ROM image file.

### toggle

Toggles active ROM image to backup ROM image.

### once\*

If specified, only the active ROM is flashed with the new image.

#### Note:

When updating the controller firmware, it is recommended not to add this parameter so that both the active and backup ROM images will be flashed.

### Noprompt

An optional parameter that suppresses the confirmation prompt.

## Examples

```
ARCCONF ROMUPDATE 1 C:\firmwareImage\SmartFW.bin noprompt
ARCCONF ROMUPDATE 1 toggle
ARCCONF ROMUPDATE 1 C:\firmwareImage\SmartFW.bin once
```

## 2.28 arccnf saveconfig

### Description

**Note:** This command is supported on all OSs that support maxView Storage Manager.

Saves the controller configuration to a XML server template file, including the controller type, operational settings, physical drive size, logical drive size, RAID level, and more. Use this file with the PLAYCONFIG command to deploy the same controller configuration to other servers in your storage space; see [arccnf playconfig](#) on page 25 for more information.

**Note:** Be sure to check the log file to verify that the configuration XML file was created successfully. The exit codes, shown below, indicate the success or failure of the operation.

Code	Value	Meaning
SUCCESS	0	Configuration XML generated successfully.
FAILURE_GENERAL	1	An error occurred and the configuration XML could not be generated.

### Syntax

```
ARCCONF SAVECONFIG [Input XML File] [LogFile] [nologs]
```

## Parameters

### Input XML File

The pathname of the server template file. The default name (if you omit this parameter) is C:\PMCS\Log\saveconfig.xml.

### LogFile

The pathname of the error log file. By default, the error log is available at C:\PMCS\Log\saveconfig.log.

## Examples

```
ARCCONF SAVECONFIG server1_config.xml C:\LOGS\SERVER1.LOG
```

## 2.29 arccnf savesupportarchive

### Description

Saves configuration and status information to help diagnose a problem with your system. Saved information includes device logs, drive logs, event logs, error logs, controller logs, history logs, basecode logs, and SMART statistics.

By default, the log files are saved in the Support folder in the standard logs directory for your operating system (/var/log for Linux, and so on).

### Syntax

```
ARCCONF SAVESUPPORTARCHIVE [Path]
[Firmware|Arccnf|Storlib|Basecode|GUI|Redfish] [nologs]
```

## Parameters

### Path

Path to store the log files.

### Log type:

One of these options:

- Firmware: saves Firmware logs
- Arccnf: saves Arccnf logs
- Storlib: saves StorLib logs
- Basecode: saves basecode logs
- GUI: saves GUI logs
- Redfish: saves Redfish logs

## Examples

```
ARCCONF SAVESUPPORTARCHIVE
ARCCONF SAVESUPPORTARCHIVE Firmware
```

## 2.30 arconf setarrayparam

### Description

Changes a parameter of an array.

### Syntax

```
ARCCONF SETARRAYPARAM <Controller#> <Array#> SPARETYPE <Type> [nologs]
ARCCONF SETARRAYPARAM <Controller#> <Array#> SSDIOBYPASS <enable/disable>
[nologs]
```

### Parameters

#### Controller#

Controller number

#### Array#

Array number to be modified

#### SPARETYPE

Sets the spare type for the array:

- 1 : Dedicated—A spare that replaces a failed drive in the array, and is shareable between arrays.
- 2: Autoreplace—A spare that replaces a failed drive in the array, and is *not* sharable between arrays.

#### SSDIOBYPASS

Enables or disables I/O bypass for all logical devices in the array. Default is enabled.

- 1: Enable—I/O bypass on array will be enabled.
- 2 : Disable—I/O bypass on array will be disabled.

### Examples

```
ARCCONF SETARRAYPARAM 1 0 SPARETYPE 1
ARCCONF SETARRAYPARAM 1 0 CONSOLIDATESPACE
ARCCONF SETARRAYPARAM 1 0 SSDIOBYPASS enable
```

## 2.31 arconf setboot

### Description

Sets the controller as a boot device for the system. This command is available only when the controller is offline.

### Syntax

```
ARCCONF SETBOOT <Controller#> LOGICALDRIVE <LogicalDrive#> [TYPE <Boot Type>]
[nologs]
ARCCONF SETBOOT <Controller#> DEVICE <Channel# ID#> TYPE <Boot Type> [nologs]
ARCCONF SETBOOT <Controller#> ENABLE [nologs]
```

### Parameters

#### Controller#

Controller number

### LogicalDrive#

Logical drive number to mark as the boot device

### Channel# ID#

Channel and ID of the physical device to mark as the boot device

### TYPE <Boot Type>

Boot type of the logical or physical device:

- Primary - Primary boot logical/physical device
- Secondary - Secondary boot logical/physical device
- None - Non-bootable

### ENABLE

Sets the controller as a boot controller

### Examples

```
ARCCONF SETBOOT 1 LOGICALDRIVE 0 TYPE primary
ARCCONF SETBOOT 1 DEVICE 0 5 TYPE secondary
ARCCONF SETBOOT 1 ENABLE
```

## 2.32 arccnf setcache

### Description

Changes the cache mode for a logical drive, or the write cache mode for all drives or a single physical drive on a controller.

### Syntax

```
ARCCONF SETCACHE <Controller#> LOGICALDRIVE <LogicalDrive#> <logical mode>
[noprompt] [nologs]
ARCCONF SETCACHE <Controller#> DRIVEWRITECACHEPOLICY <DriveType> <CachePolicy>
[noprompt] [nologs]
ARCCONF SETCACHE <Controller#> CACHERATIO <read#> <write#> [nologs]
ARCCONF SETCACHE <Controller#> WAITFORCACHEROOM <enable | disable> [nologs]
ARCCONF SETCACHE <Controller#> NOBATTERYWRITECACHE <enable | disable> [nologs]
ARCCONF SETCACHE <Controller#> WRITECACHEBYPASSTHRESHOLD <threshold size>
[nologs]
ARCCONF SETCACHE <Controller#> RECOVERCACHEMODULE [nologs]
```

### Parameters

#### Controller#

The controller number

#### LogicalDrive#

The number of the logical drive whose cache will be altered

#### Logical mode

Logical drive cache mode:

- con—cache enabled
- coff—cache disabled

#### Channel/ID

Lists the space-delimited channel number and device number pairs for each device.

#### DRIVETYPE

- Configured—drive write cache policy for configured drives
- Unconfigured—drive write cache policy for unconfigured drives

### CachePolicy

Write cache policy setting; choose any of the following values

- 0-default
- 1-enable
- 2-disable

### CACHERATIO <read#> <write#>

Sets the cache ratio for the controller:

- read#—read cache percentage
- write#—write cache percentage

### WAITFORCACHEROOM

Wait for room in the read/write cache when full instead of automatically bypassing it in favor of higher performance. Enabling this feature prevents RAID 1 inconsistencies that occur whenever the host changes buffer contents during write operations.

- Enable—wait for room in the read/write cache
- Disable—do not wait for room in the read/write cache

### NOBATTERYWRITECACHE

Enables write caching when a battery or supercapacitor is not present or fully charged. This setting applies to all logical drives on the controller; at least one logical drive must exist before usage.

- Enable—enable write caching on controller without fully charged battery or supercapacitor
- Disable—disable write caching on controller without fully charged battery or supercapacitor

**Caution:** Enabling write caching without a fully charged battery/supercapacitor may cause data loss in the event of a power failure.

### WRITECACHEBYPASSTHRESHOLD

Sets the write cache bypass threshold for the controller. This allows you to choose a value to bypass the cache when the large write reaches that threshold. Units are in KB and the value must be a multiple of 16 KB. The target can be any valid controller.

- Threshold size—the valid threshold size is between 16 KB and 1040 KB.

### RECOVERCACHEMODULE

Recovers the failed cache module.

### Examples

```
ARCCONF SETCACHE 1 CACHERATIO 60 40
ARCCONF SETCACHE 1 WAITFORCACHEROOM enable
ARCCONF SETCACHE 1 NOBATTERYWRITECACHE enable
ARCCONF SETCACHE 1 DRIVEWRITECACHEPOLICY Configured 0 Unconfigured 1 hba 2
ARCCONF SETCACHE 1 WRITECACHEBYPASSTHRESHOLD 1040
ARCCONF SETCACHE 1 RECOVERCACHEMODULE
```

## 2.33 arccnf setconf

### Description

Resets the controller configuration. Logical drives are deleted, hard disks are reset to the READY state, cache contents are lost, and controller settings are reset to default values. Optionally, you can clear the maxCrypto configuration, including all keys, passwords, and maxCrypto users (administrator and standard user).



## Syntax

```
SETCONFIG <Controller#> <DEFAULT | CLEARMAXCRYPTOCONFIG> [noprompt] [nologs]
```

## Parameters

### Controller#

The controller number

### Default

Restores the controller's default configuration.

### Clearmaxcryptoconfig

Restores the default maxCrypto configuration.

### Noprompt

No prompt for confirmation.

## Examples

```
ARCCONF SETCONFIG 1 DEFAULT
ARCCONF SETCONFIG 1 CLEARMAXCRYPTOCONFIG
```

## 2.34 arccnf setconnectormode

### Description

Use this command to configure controller connectors to different operating modes:

- HBA Mode—Allows the connector to act and be used as a Host Bus Adapter. RAID functions of the controller are disabled. All attached drives are surfaced as RAW devices.
- RAID: Hide RAW—All RAID functions of the controller are enabled for the connector, but RAW devices are not exposed to the operating system.
- Mixed—RAID volumes and RAW drives on the connector are exposed to operating system.

### Syntax:

```
ARCCONF SETCONNECTORMODE <Controller#> <Connector #> <Functional Mode#>
<Connector #> <Functional Mode#> ... [noprompt] [nologs]
```

## Parameters

### Controller#

Controller number.

### Connector#

Connector number.

### Functional Mode#

One of the following values:

- 1 - HBA Mode
- 2 - RAID: Hide RAW
- 3 - Mixed

### nologs

Suppresses log output.

### Examples

```
ARCCONF SETCONNECTORMODE 1 1 1
ARCCONF SETCONNECTORMODE 1 3 3
```

## 2.35 **arccnf setcontrollermode**

### Description

Use this command to configure the controller operating mode for all connectors:

- HBA Mode—Allows the controller to act and be used as a Host Bus Adapter. RAID functions of the controller are disabled. All attached drives are surfaced as RAW devices.
- RAID: Hide RAW—All RAID functions of the controller are enabled, but RAW devices are not exposed to the operating system.
- Mixed—RAID volumes and RAW drives are exposed to operating system.

### Syntax

```
ARCCONF SETCONTROLLERMODE <Controller#> <Controller Mode> [nologs]
```

### Parameters

#### Controller Mode

Change a controller's mode.

- 2 - HBA Mode
- 3 - RAID: Hide RAW
- 5 - Mixed

### Examples

```
ARCCONF SETCONTROLLERMODE 1 2
```

## 2.36 **arccnf setcontrollerparam**

### Description

Changes a parameter of a controller.

## Syntax

```

ARCCONF SETCONTROLLERPARAM <Controller#> QUEUEDEPTH <QDepth> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> SPAREACTIVATIONMODE <Mode> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> ELEVATORSORT <Enable | Disable>
[nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> LATENCY <Latency> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> SANITIZELOCK <sanitizeLock> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> MIXEDVOLUMES <Enable | Disable>
USERROLE <userrole> PASSWORD <password> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> FWLOCK <Enable | Disable> USERROLE
<userrole> PASSWORD <password> [nologs]

ARCCONF SETCONTROLLERPARAM <Controller#> OOBINTERFACE <OOBinterface> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> OOBINTERFACE PBSI I2CADDRESS <i2cAddress>
I2CCLOCKSPPEED
<i2cClockSpeed> I2CCLOCKSTRETCH <i2cClockStretch> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> OOBINTERFACE MCTP I2CADDRESS <i2cAddress>
SMBUSDEVICETYPE
<SMBusDeviceType> SMBUSCHANNEL <SMBusChannel> STATICEID <StaticEID> VDMNOTIFICATION
<VDMNotification> [nologs]

ARCCONF SETCONTROLLERPARAM <Controller#> DISCOVERYPROTOCOL <All discoveryProtocol>
[noprompt] [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> DISCOVERYPROTOCOL <Connector# discoveryProtocol>
[Connector# discoveryProtocol]... noprompt] [nologs]

ARCCONF SETCONTROLLERPARAM <Controller#> EXPANDERSCANDURATION <duration> [nologs]

```

## Parameters

### Controller#

Controller number

### QUEUEDEPTH <QDepth>

Sets the queue depth for the controller. Valid values are 0, 2, 4, 8, 16, and 32. A value of 0 indicates automatic queue depth.

### SPAREACTIVATIONMODE <mode>

Sets the spare activation mode from activation on failure to predictive spare activation. Valid values are:

- 0 : Activate on failure (default)
- 1 : Activate on predictive failure

### ELEVATORSORT

Sets the behavior of the controller cache write Elevator sort algorithm.

### LATENCY

Sets the flexible latency scheduler. Valid values are:

- 0 : Disable (default).
- 1 : Low. Sets value to 250.
- 2 : Medium. Sets value to 100.
- 3: High. Sets value to 50.
- 4: Aggressive level 1. Sets value to 30.
- 5: Aggressive level 2. Sets value to 10.

### SANITIZELOCK

Sets the Sanitize lock on the controller.

- sanitizeLock
  - None - Default setting
  - Freeze - Freezes the Sanitize operation on all supported drives

- AntiFreeze - Blocks setting the Freeze mode on all supported drives. Prevents further attempts to freeze the Sanitize operation on the hard drive.

### MIXEDVOLUMES

Enables mixing of encrypted and plaintext logical devices. Valid values are:

- Enable: Authorized users have the option to create encrypted logical devices or plaintext logical devices (not encrypted).
- Disable: New logical devices will be encrypted, with no option to create plaintext logical devices.

### FWLOCK

Locks/unlocks controller firmware update. Valid values are:

- Enable: Authorized users can update the controller firmware.
- Disable: Controller firmware cannot be updated.

### USERROLE <userrole> PASSWORD <password>

maxCrypto user-role and password. Valid values are:

- crypto (maxCrypto administrator)
- user (standard user)

### OOBINTERFACE

Sets the Out-of-band interface settings for the controller. Parameters for configuring OOB interface:

- PBSI
- MCTP

### PBSI

Sets the Out-of-band interface as PBSI.

PBSI Parameters	Description
I2CADDRESS	Sets the I2C Address of the controller. Hexadecimal input from range of 0x00—0xFF
i2cClockSpeed	Sets I2C clock speed <ul style="list-style-type: none"> <li>• 0 : I2C clock speed disable (Default)</li> <li>• 2 : I2C clock speed 100 kHz</li> <li>• 3 : I2C clock speed 400 kHz</li> </ul>
i2cClockStretch	Sets the I2C clock stretch <ul style="list-style-type: none"> <li>• Enable : Enables I2C Clock Stretching</li> <li>• Disable : Disables I2C Clock Stretching</li> </ul>

### MCTP

Sets the Out-of-band interface as MCTP.

MCTP parameters	Description
I2CADDRESS	Sets the I2C address of the controller. Hexadecimal input. For valid range, refer to the Management Component Transport Protocol (MCTP) SMBus/I2C Transport Binding Specification document.
SMBusDeviceType	Sets System Management(SM) Bus Device Type <ul style="list-style-type: none"> <li>• 0 : Default</li> <li>• 1 : Fixed: Fixed and not discoverable. If this value is set, the I2C address is fixed and will not be able to be assigned with ARP.</li> <li>• 2 : Address Resolution Protocol (ARP) capable Bus master can assign the I2C address dynamically to slave devices to resolve the address conflicts.</li> </ul>
SMBusChannel	Sets the System Management(SM) Bus Channel setting <ul style="list-style-type: none"> <li>• Enable : Enables SM Bus Channel</li> <li>• Disable : Disables SM Bus Channel</li> </ul>
StaticEID	Sets the Static End Point Identifier (EID) Setting <ul style="list-style-type: none"> <li>• Enable : Enables Static EID</li> <li>• Disable : Firmware passes EID as 0 (internally) and the bus owner assigns the EID.</li> </ul>

MCTP parameters	Description
VDMNotification	Sets the Vendor Defined Message (VDM) discovery notification <ul style="list-style-type: none"> <li>• Enable : Enabling VDM Discovery Notify supports sending discovery notification during a PCI bus re-enumeration. This message is available for use as a common message for enabling an endpoint to announce its presence to the bus owner. This is typically used as a part of the endpoint discovery process when an MCTP device is hot-plugged onto or becomes powered-up on an MCTP bus.</li> <li>• Disable : Disables VDM discovery Notification</li> </ul>

## DISABLE

Sets the Out-of-band interface as Disable

## DISCOVERYPROTOCOL

Sets the protocol of a connector to discover the connected backplane. Available connector discovery protocols are: UBM and SGPIO.

- All: Sets the discovery protocol for all the connectors
- DiscoveryProtocol:
  - UBM: The controller uses the UBM protocol to communicate with the backplane attached to the connector.
  - SGPIO: The controller uses SGPIO to communicate with the backplane attached to the connector.

## EXPANDERSCANDURATION

Expander Minimum Scan Duration: Controller waits for the specified seconds to scans/discover the drives attached to the expander on the next power cycle. Set this to a non-zero value if some devices do not appear in the topology after controller boot or rescan requests.

- duration: Time duration a controller will wait for during expander discovery. Time duration should be in seconds.

## RESET

Resets the Out-of-band interface

## Examples

```
ARCCONF SETCONTROLLERPARAM 1 SANITIZELOCK Freeze
ARCCONF SETCONTROLLERPARAM 1 MIXEDVOLUMES enable USERROLE crypto PASSWORD Abc@1234
ARCCONF SETCONTROLLERPARAM 1 FWLOCK enable USERROLE crypto PASSWORD Abc@1234
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE Disable
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE PBSI I2CAddress 0xD0 I2CCLOCKSPEED 3
I2CCLOCKSTRETCH Enable
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE MCTP I2CAddress 0xD0 SMBUSDEVICETYPE 1
SMBUSCHANNEL Enable STACEID Enable VDMNotification Enable
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE Reset
ARCCONF SETCONTROLLERPARAM 1 DISCOVERYPROTOCOL ALL UBM
ARCCONF SETCONTROLLERPARAM 1 DISCOVERYPROTOCOL 0 SGPIO 1 UBM 2 SGPIO
ARCCONF SETCONTROLLERPARAM 1 EXPANDERSCANDURATION 5
```

## 2.37 arconf setmaxcache

### Description

Updates the maxCache write cache policy for one or more logical drives.

**Syntax: Write Caching**

```
ARCCONF SETMAXCACHE <Controller#> LOGICALDRIVE <LD#> [<LD#> <LD#>..]
MAXCACHEWRITEPOLICY <Policy> [nologs]
```

**Parameters****Controller#**

The controller number.

**LogicalDrive#**

The number of the logical drive. You can specify one or more logical drives.

**MAXCACHEWRITEPOLICY**

Changes the MaxCache Write Cache policy on the logical device.

- Policy - MaxCache Write Cache Policy
- WB - Write Back Cache Policy
- WT - Write Through Policy

**Examples**

```
ARCCONF SETMAXCACHE 1 LOGICALDRIVE 0 MAXCACHEWRITEPOLICY wt
```

**2.38 arccnf setname****Description**

Renames a logical drive.

**Syntax**

```
ARCCONF SETNAME <Controller#> LOGICALDRIVE <LogicalDrive#> <New Name> [nologs]
```

**Parameters****Controller#**

Controller number

**LogicalDrive#**

The number of the logical drive to be renamed

**New Name**

The new name of the logical drive, and must be of 1 to 64 characters long, using only ASCII characters.

**Examples**

```
ARCCONF SETNAME 1 LOGICALDRIVE 1 BACKUP_A
```

## 2.39 arconf setperform

### Description

Changes controller settings based on the application type.

### Syntax

```
ARCCONF SETPERFORM <Controller#> MNPDELAY <Delay> [nologs]
ARCCONF SETPERFORM <Controller#> DPO <Enable | Disable> [nologs]
```

### Parameters

#### Controller#

The controller number

#### MNPDELAY <Delay>

Sets the monitor and performance delay for the controller, in seconds. Default is 60 minutes (3600 seconds).

#### DPO

Enables or disables the degraded performance setting for the controller. Default is disabled.

### Examples

```
ARCCONF SETPERFORM 1 MNPDELAY 1800
ARCCONF SETPERFORM 1 DPO enable
```

## 2.40 arconf setpower

### Description

Modifies the power management settings.

### Syntax

```
ARCCONF SETPOWER <Controller#> POWERMODE <mode> SURVIVALMODE <mode> [nologs]
```

### Parameters

#### Controller#

The controller number.

#### POWERMODE

Specifies the power mode for the controller.

- 1: Minimum power-Set static settings to lowest possible values and reduce power dynamically based on workload.
- 2: Balanced power-Set static settings based on configuration and reduce power dynamically based on workload.
- 3: Maximum performance-Set static settings to highest possible values and do not reduce power dynamically.

## SURVIVALMODE

Survival mode allows the controller to throttle back dynamic power settings to their minimum when temperatures exceed the warning threshold.

This allows the server to continue running in more situations, but performance may decrease.

- Enable-Survival mode enabled.
- Disable-Survival mode disabled.

### Examples

```
ARCCONF SETPOWER 1 POWERMODE 2
ARCCONF SETPOWER 1 SURVIVALMODE 1
```

## 2.41 arconf setpriority

### Description

Changes a task's execution priority or a controller's global background task priority.

### Syntax

```
ARCCONF SETPRIORITY <Controller#> <REBUILD|EXPAND> <New Priority> [nologs]
```

### Parameters

#### Controller#

The controller number

#### New Priority

LOW, MEDIUM, or HIGH. For REBUILD only: MEDIUMHIGH (if rapid rebuild priority is supported on the controller).

#### REBUILD

Sets the controller's rebuild priority.

#### EXPAND

Sets the controller's capacity expansion (OCE) priority.

### Examples

```
ARCCONF SETPRIORITY 1 EXPAND LOW
SETPRIORITY 1 REBUILD MEDIUM
```

## 2.42 arconf setstate

### Description

Changes the state of a physical device or logical device or maxcache from its current state to the designated state.



## Syntax

```
ARCCONF SETSTATE <Controller#> DEVICE <Channel#> <Device#> <State> [ARRAY <AR#>
[AR#] ... ][SPARETYPE <TYPE>][noprompt] [nologs]
ARCCONF SETSTATE <Controller#> LOGICALDRIVE <LD#> OPTIMAL [ADVANCED <option>]
[noprompt] [nologs]
ARCCONF SETSTATE <Controller#> MAXCACHE <LD#> OPTIMAL [noprompt] [nologs]
```

## Parameters

### Controller#

The controller number

### Channel#

The channel number for the drive.

### Device#

Device number for the device.

### LD#

Logical drive number.

### AR#

Array number.

### State

- HSP—Create a hot spare from a ready drive. Dedicates the HSP to one or more .
- RDY—Remove a hot spare designation. Attempts to change a drive from Failed to Ready.
- DDD—Force a drive offline (to Failed).
- EED—Enable the erased drive.

### MAXCACHE

Optional keyword for maxCache devices only. Include if State is HSP or RDY, and the hot spare is for a maxCache device.

- maxCache Id#—maxCache logical device ID to be forced optimal.
- Type
  - 1: Dedicated—A dedicated spare temporarily takes over for a failed drive and can be shared between arrays.
  - 2: Autoreplace—An autoreplace spare replaces a failed drive and cannot be shared between arrays.

### SPARETYPE

Sets the sparetype for the array.

### Noprompt:

No prompt for confirmation.

## Examples

```
ARCCONF SETSTATE 1 LOGICALDRIVE 1 OPTIMAL
ARCCONF SETSTATE 1 DEVICE 0 0 DDD
ARCCONF SETSTATE 1 DEVICE 0 0 RDY
ARCCONF SETSTATE 1 DEVICE 0 0 HSP ARRAY 0
ARCCONF SETSTATE 1 MAXCACHE 0 OPTIMAL
```

## 2.43 **arcconf setstatsdatacollection**

### Description

Enables or disables statistics collection for a controller. To display the statistics, see [arcconf getlogs](#) on page 13 .

### Syntax

```
ARCCONF SETSTATSDATACOLLECTION <Controller#> Enable|Disable [nologs]
```

### Parameters

#### Controller#

The controller number

#### Enable

Turns statistics collection on.

#### Disable

Turns statistics collection off.

### Examples

```
ARCCONF SETSTATSDATACOLLECTION 1 ENABLE
```

## 2.44 **arcconf slotconfig**

### Description

Lists the channel ID and device ID of the devices in each slot of an enclosure. A slot with no devices marked as EMPTY.

### Syntax

```
ARCCONF SLOTCONFIG <Controller#> <EnclosureID#> <Slot#> [nologs]
ARCCONF SLOTCONFIG <Controller#> <EnclosureID#> MAP [nologs]
```

### Parameters

#### Controller#

The controller for which slot configuration is required.

#### EnclosureID#

The enclosure where slot configuration is required.

#### Slot#

The slot number of the drive where information is required.

#### MAP

Display the slot configuration of an enclosure.

## Examples

```
ARCCONF SLOTCONFIG 1 0 2 2
ARCCONF SLOTCONFIG 1 CN0 1 0
ARCCONF SLOTCONFIG 1 CN0 2 MAP
ARCCONF SLOTCONFIG 1 ALL MAP
```

## 2.45 arccnf smp

### Description

Sends a SAS Management Protocol (SMP) function request to a SMP target device.

### Syntax:

```
ARCCONF SMP <Controller#> Enclosure <Connector# Channel# Device#> Expander
<Expander#> <CommandType1> [ASCII] [nologs]
ARCCONF SMP <Controller#> Enclosure <Connector# Channel# Device#> Expander
<Expander#> <CommandType2> PHY <PHY#> [ASCII] [nologs]
```

### Parameters

#### Controller#

Controller number.

#### Connector# Channel# ID#

Connector ID, Channel ID and Device ID of the enclosure that contains the expander.

#### Expander#

Expander number on the controller (SMP target device).

#### PHY#

The PHY Identifier (valid only for Discover and PHY Error Log Request).

#### CommandType#

CommandType1:

- RGR - Report General Request
- RMR - Report Manufacturer Request

CommandType2:

- DR - Discover Request
- RPELR - Report PHY Error Log Request

#### ASCII

Displays the SMP response in ASCII format along with Hex formatted output.

### Examples

```
ARCCONF SMP 1 Enclosure 1 2 0 Expander 0 RGR
ARCCONF SMP 1 Enclosure 1 2 0 Expander 1 DR 0
```

## 2.46 arccnf splitmirror

### Description

Splits an array consisting of one or more RAID 1, RAID 10, RAID 1(ADM) or RAID10(ADM) logical devices into two new arrays with identical contents.

### Syntax

```
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> SPLITWITHBACKUP [nologs]
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> REMIRROR [nologs]
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> ROLLBACK [nologs]
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> ACTIVATEBACKUP [nologs]
```

### Parameters

#### Controller#

Controller number

#### Array#

Array number

#### SPLITWITHBACKUP

Splits the array into two new arrays: a primary array and a backup array, with the following characteristics:

- If the original array contained RAID 1 or RAID 10 drives, the primary array will contain RAID 0 drives.
- If the original array contained RAID 1(ADM) drives, the primary array will contain RAID 1 drives.
- If the original array contained RAID 10 (ADM) drives, the primary array will contain RAID 1+0 drives.

The backup array always contains RAID 0 logical drives. The primary array continues to be fully accessible to the operating system while the backup array is hidden from the operating system.

#### REMIRROR

Remirrors the array by preserving the existing data and discarding the backup array. This option re-creates the original mirrored array with the contents of the primary array.

#### ROLLBACK

Remirrors the array by rolling back to the contents of the backup array and discarding existing data. This option re-creates the mirrored array but restores its contents to the point in time when the backup array was created.

**Caution:** We do not recommend using this option while the array is online, or while the logical drive to be rolled back is mounted or in use by the operating system.

#### ACTIVATEBACKUP

Activates the backup array and makes it fully accessible to the operating system.

### Examples

```
ARCCONF SPLITMIRROR 1 ARRAY 0 SPLITWITHBACKUP
ARCCONF SPLITMIRROR 1 ARRAY 0 REMIRROR
ARCCONF SPLITMIRROR 1 ARRAY 0 ROLLBACK
ARCCONF SPLITMIRROR 1 ARRAY 0 ACTIVATEBACKUP
```

## 2.47 arccconf task

### Description

Performs a task on a logical drive, physical drive, array, or maxCache logical device. Uninitializes physical drives on a controller. Erases an encrypted logical drive or array, encodes (encrypts) a plaintext logical drive, and creates a new key for an encrypted logical device.

### Syntax:

```
ARCCONF TASK
ARCCONF TASK START <Controller#> DEVICE <Channel# ID#> <task> [PATTERN
<erasePattern>][noprompt] [nologs]
ARCCONF TASK START <Controller#> DEVICE ALL UNINITIALIZE [nologs]
ARCCONF TASK STOP <Controller#> DEVICE <Channel#> <ID#> [nologs]
```

### Syntax: maxCrypto Usage

```
ARCCONF TASK
ARCCONF TASK START <Controller#> LOGICALDRIVE <LogicalDrive#> CRYPTOERASE
USERROLE <userrole> PASSWORD <password>
ARCCONF TASK START <Controller#> ARRAY <Array#> CRYPTOERASE USERROLE <userrole>
PASSWORD <password>
ARCCONF TASK START <Controller#> LOGICALDRIVE <LogicalDrive#> ENCODE DATA
<Preserve | Discard> USERROLE <userrole> PASSWORD <password>
ARCCONF TASK START <Controller#> LOGICALDRIVE <LogicalDrive#> REKEY USERROLE
<userrole> PASSWORD <password>
ARCCONF TASK START <Controller#> ARRAY <Array#> REKEY USERROLE <userrole>
PASSWORD <password>
```

### Parameters

#### Controller#

The controller number

#### Channel# ID#

Channel number and device ID for the device

#### Options:

- Physical device options:
  - `secureerase [password] [PATTERN <pattern>]`—removes all data from the drive in a secure fashion to prevent any possible recovery of the erased data. Erase patterns:
    - 1 : Zero - Initializes all blocks to zero.
    - 2 : Random Zero - This erase method writes random data on the drive for the first pass then writes zeros for the second pass.
    - 3 : Random Random Zero - This erase method writes random data on the drive for the first and second passes and then zeros for the third pass.
    - 4 : Reserved.
    - 5 : Block Erase Sanitize Method - SSDs only. Erase voltage is applied to all NAND cells.
    - 6 : Overwrite Sanitize Method - HDDs only. Initializes blocks using complex multi-byte data pattern.
  - `Unrestricted`—With the Sanitize Erase option, the physical device is available for configuration if sanitize erase fails or could not complete. If not provided, value defaults to 'Restricted'. With the default option, if Sanitize Erase fails, the only operation allowed is to start another sanitize.
  - `UNINITIALIZE`—When specified with ALL, clears meta-data and any OS partitions from all drives on the controller; existing data on the drive is destroyed.

**Cryptoerase**

Erases an encrypted logical drive or array. (After erasing, the logical device remains encrypted.)

**Encode Data <Preserve | Discard>**

Encrypts a logical drive or array, with option of preserving or discarding the original data.

**Rekey**

Generates a new key for encrypted devices.

**USERROLE <userrole> PASSWORD <password>**

maxCrypto user-role and password. Valid values are:

- crypto (maxCrypto administrator)
- user (standard user)

**Examples**

```
ARCCONF TASK START 1 DEVICE 0 0 SECUREERASE PATTERN 1
ARCCONF TASK STOP 1 DEVICE 0 0
ARCCONF TASK START 1 DEVICE ALL UNINITIALIZE
ARCCONF TASK START 1 LOGICALDRIVE 0 CRYPTOERASE USERROLE crypto password Abc@1234
ARCCONF TASK START 1 LOGICALDRIVE 0 ENCODE DATA preserve USERROLE crypto password Abc@1234
ARCCONF TASK START 1 ARRAY 0 REKEY USERROLE crypto password Abc@1234
```

**2.48 arccnf uninit****Description**

Uninitializes one or more physical drives. The uninitialize command clears meta-data and any OS partitions from a drive; existing data on the drive is destroyed.

**Note:** Uninitialized drives are compatible with any HBA and can be exchanged with drives on the motherboard's SATA interface.

**Syntax:**

```
ARCCONF UNINIT <Controller#> <Channel# Drive#> [Channel# Drive#] ... [nologs]
ARCCONF UNINIT <Controller#> ALL [nologs]
```

**Parameters****Controller#**

Controller number.

**Channel#**

The channel number of the device to be uninitialized.

**Drive#**

The drive number of the device to be uninitialized.

**ALL**

Uninitializes all physical devices on the controller.

**nologs**

Suppresses log output for the command.

## Examples

```
ARCCONF UNINIT 1 0 12 0 13  
ARCCONF UNINIT 1 ALL
```

## A Running ARCCONF in the UEFI Shell

This appendix describes how to run ARCCONF in the UEFI shell. UEFI/ARCCONF supports a subset of commands available on the command line. Most commands have the same form and syntax as their command line counterparts, with the exceptions noted below in [UEFI/ARCCONF Commands](#) on page 48. Additionally, some commands are supported in UEFI/ARCCONF only.

### Prerequisites

To run UEFI/ARCCONF, ensure that your system meets these requirements:

- System is running UEFI Shell v2.2 or higher
- MSCC UEFI driver is installed:
  1. Boot the machine to the UEFI shell prompt.
  2. Type: `drivers`
  3. Verify that 'MSCC UEFI Driver (version)' is listed.

### Starting UEFI/ARCCONF

To start UEFI/ARCCONF:

1. Boot the machine to the UEFI shell prompt.
2. At the prompt, enter a command in the form:

```
arccnf <command_name> <parameters> ...
```

3. To see a list of supported commands, type `ARCCONF` at the prompt; to include pagebreaks, type `ARCCONF -b`. For help with a specific command, type `ARCCONF <command_name> help`.

### UEFI/ARCCONF Commands

The table below lists the commands supported in UEFI/ARCCONF. Follow the link in the **Usage** column for command forms and syntax. Where syntax differs from the command line, a separate usage statement is listed.

**Table 4 • UEFI/ARCCONF Commands**

Command	Description	Usage
CONSISTENCYCHECK	Toggles background consistency check modes of the controller. Usage : CONSISTENCYCHECK <Controller#> <on [Delay]   off> [noprompt] Usage : CONSISTENCYCHECK <Controller#> PARALLELCOUNT <Count> Usage : CONSISTENCYCHECK <Controller#> INCONSISTENCYREPAIRPOLICY <Enable   Disable> see <a href="#">arccnf consistencycheck</a> on page 5	
CREATE	Creates a new logical drive; optionally, enables logical drive read caching, write caching.	see <a href="#">arccnf create</a> on page 6
DELETE	Deletes a logical drive.	see <a href="#">arccnf delete</a> on page 8
EXPANDERLIST	Returns list of disk drive expanders on a controller.	see <a href="#">arccnf expanderlist</a> on page 9



Command	Description	Usage
EXPANDERUPGRADE	Allows new firmware to be flashed to an enclosure or expander.	see <a href="#">arccconf expanderupgrade</a> on page 10
GETCONFIG	Lists information about controllers and physical drives.	see <a href="#">arccconf getconfig</a> on page 11
GETLOGS	Provides access to controller status, event logs, and usage statistics. Usage : GETLOGS <Controller#> <Type 1> Usage : GETLOGS <Controller> <Type 2> LOGICALDRIVE <LD#> see <a href="#">arccconf getlogs</a> on page 13	
GETVERSION	Lists version information for all controllers or a specific controller's software components.	see <a href="#">arccconf getversion</a> on page 14
IDENTIFY	Identifies a physical device by blinking its LEDs.	see <a href="#">arccconf identify</a> on page 15
IMAGEUPDATE	Allows new firmware to be flashed to the hard drive.	see <a href="#">arccconf imageupdate</a> on page 16
LIST	Lists controllers in the system.	see <a href="#">arccconf list</a> on page 17
PASSTHROUGH	Sends SCSI passthrough protocol (SCSI) function request to a disk drive.  <b>Note:</b> This command is supported in UEFI/ARCCONF only.	see <a href="#">arccconf passthrough</a> on page 22
RESCAN	Check for removal of disk drives and for connection of new disk drives to the controller.	see <a href="#">arccconf rescan</a> on page 26
ROMUPDATE	Allows new firmware and BIOS to be flashed to the controller. Usage : ROMUPDATE <Controller#> <ImagePath> [noprompt] [nologs] see <a href="#">arccconf romupdate</a> on page 27	
SAVESUPPORTARCHIVE	Saves configuration and status information. Usage : SAVESUPPORTARCHIVE [Path] see <a href="#">arccconf savesupportarchive</a> on page 29	
SETARRAYPARAM	Changes a parameter of an array.	see <a href="#">arccconf setarrayparam</a> on page 30
SETBOOT	Sets controller as a boot device for the system.	see <a href="#">arccconf setboot</a> on page 30
SETCONFIG	Resets the controller configuration. Usage : SETCONFIG <Controller#> <DEFAULT> [noprompt] [nologs] see <a href="#">arccconf setconfig</a> on page 32	
SETCONNECTORMODE	Change the connector operating mode.	see <a href="#">arccconf setconnectormode</a> on page 33
SETSTATE	Changes the state of a physical device or logical device.	see <a href="#">arccconf setstate</a> on page 40
SLOTCONFIG	Lists the channel ID and device ID of the devices in each enclosure slot. . A slot with no device is marked as EMPTY.  <b>Note:</b> This command is supported in UEFI/ARCCONF only.	see <a href="#">arccconf slotconfig</a> on page 42

Command	Description	Usage
SMP	Sends a SAS Management Protocol (SMP) function request to a SMP target device.	see <a href="#">arccconf smp</a> on page 43

## B Controller, Array, Logical and Physical Device Properties

This appendix provides detailed description of the controller, array, logical and physical device properties of ARCCONF in the command line.

**Table 5 • Controller**

Properties	Description
Controller Status	Overall status of the controller based on its resources
Controller Mode	Operating mode of the controller
Channel description	Channel description of the controller
Controller Model	Model name of the controller connected to the system
Controller Serial Number	A unique number assigned to the controller, used for identification and inventory purposes
Controller World Wide Name	A World Wide Name (WWN) is a unique identifier of the controller
Physical Slot	PCI slot number to which the controller is connected
Temperature	Current temperature of the controller
Host bus type	Type of host expansion bus standard
Host bus speed	Speed of the host expansion bus standard in MHz
Host bus link width	Data width of the host expansion bus standard in bits
PCI Address (Bus:Device:Function)	PCI address describes the PCI address for the controller
Number of Ports	Describes number of internal and external ports of the controller
Internal Port Count	Describes number of internal ports of the controller
External Port Count	Describes number of external ports of the controller
Defunct disk drive count	Number of failed drive
NCQ status	Native Command Queuing, or NCQ, lets SATA disk drives arrange commands into the most efficient order for optimum performance
Performance Mode	To ensure optimal performance of the logical drives in your storage space, you can select an application-specific Performance mode to improve I/O throughput based on the needs of your application
Queue Depth	Queue Depth controls the behavior of the cache write queue. This option is used to tune controller performance for video applications. The valid values are 2, 4, 8, 16, 32, or Automatic
Monitor and Performance Delay	This option controls the behavior of the controller Monitor and Performance Analysis Delay and is expressed in values ranging from 0 to 1440. This option is primarily used to tune the controller's performance for video applications.

Properties	Description
Elevator Sort	This option controls the behavior of the controller cache write elevator sort algorithm. This option is used to tune controller performance for video applications. The possible options are Enable or Disable.
Degraded Mode Performance Optimization	Degraded Mode Performance Optimization setting applies to RAID 5 logical devices in Degraded mode only. Enabling this setting directs the controller to improve performance of large read requests by buffering physical drive requests. Disabling this feature forces the controller to read from the same drives multiple times. This option is used to tune controller performance for video applications. The possible options are Enable or Disable.
Latency	Latency describes Flexible Latency Schedule (FLS) setting. Flexible Latency Scheduler (FLS) is a controller option where the controller can re-prioritize I/O requests to prevent some requests to HDDs from timing out. Under normal operation (when FLS is disabled, or in controllers that don't support FLS), the controller will sort incoming requests in order to minimize the amount of travel for the HDD's read heads (Elevator Sort). This strategy works well for workloads that access sequential data, or workloads that require multiple requests from localized sectors in the drive. For highly random workloads, such as transaction processing, some requests will end up on the wrong side of the disk platter and, due to their high latency, will be marked as timed out. When FLS is enabled, it will detect these high-latency requests and apply a cut-off value, after which it will suspend elevator sorting and service the request right away
Post Prompt Timeout	Post prompt timeout describes the F1/F2 POST prompt timeout for the controller during system boot
Statistics Data Collection Mode	Use the Statistics Viewer to view the advanced statistics for the controllers, hard drives, SSDs, logical drives, and maxCache devices in your storage space, including virtual volumes and the maxCache container
Boot Controller	Boot controller can be changed only in the offline environment and only applies to booting in Legacy BIOS Boot mode. The boot controller is the first controller that the system examines (after power up) to find a bootable logical device or a physical device. The controller boots from the first local drive by default. It is recommended that leaving the default setting disabled to prevent an error when booting after replacing the boot volume or controller
Primary Boot Volume	Describes which logical device or physical device is the primary boot volume on the current controller
Secondary Boot Volume	Describes which logical device or physical device is the secondary boot volume on the current controller
Driver Name	Driver name describes the name of the driver
Driver Supports SSD I/O Bypass	Driver supports SSD I/O Bypass describes whether the driver supports the SSD I/O Bypass feature
Manufacturing Part Number	Describes the hardware part number information about the controller
Manufacturing Spare Part Number	Describes the hardware spare part number information about the controller
Manufacturing Wellness Log	Describes the hardware wellness log information about the controller
NVRAM Checksum Status	Describes the NVRAM Checksum status
Sanitize Lock Setting	Set the sanitize lock policy of the controller. This policy will be applied to all SATA physical devices that support the feature. <ol style="list-style-type: none"> <li>None: No freeze lock or anti-freeze lock commands are sent to any physical device.</li> <li>Freeze: Supported physical devices are freeze locked and sanitize is not allowed.</li> </ol>

Properties	Description
	<b>3.</b> 3. Anti-Freeze: Supported physical devices are anti-freeze locked and freezing the physical devices is not allowed.
Pending Sanitize Lock Setting	Sanitize lock is in pending state, reboot the system and requires all physical devices to be power cycled or hot-plugged for the lock state to be applied to the physical devices.
Verify Write Setting	Verify write operation helps to ensure data is written to the disk's media and is verified for correctness immediately. Most importantly, the verification process helps to detect any unrecoverable error right away.
Current Power Mode	Power mode setting that determines controller static settings based on workload
Pending Power Mode	This setting determines controller static settings based on workload to be applied after reboot.
Survival Mode	Enabling survival mode will allow the controller to throttle back dynamic power settings to their minimum when temperature exceeds the warning threshold. This allows the server to continue running in more situations, but performance may decrease.
I2C Address	I2C address describes the Inter-Integrated Circuit(I2C) slave address.
I2C Clock Speed	I2C clock speed describes the Inter-Integrated Circuit(I2C) clock speed.
I2C Clock Stretching	I2C stretching status describes the Inter-Integrated Circuit(I2C) clock status.
Cache Status	Determines the preservation status of the cache module.
Cache Serial Number	A unique number assigned to the cache module
Cache memory	Cache memory size in the controller
Cache Ratio	The controller cache ratio setting determines the controller ability to adjust the amount of memory for read-ahead cache versus write cache
No-Battery Write Cache	This setting decides the controller to enable write cache when no battery is present or when the battery fails. Values are Enable or Disable.
Wait for Cache Room	This setting causes the controller to always wait for room in the read/write cache when full, instead of automatically bypassing it in favor of higher performance
Write Cache Bypass Threshold Size	All writes larger than the specified value will bypass the write cache and be written directly to the disk for non-parity RAID volumes.
Global Physical Device Write Cache Policy	Global Physical Devices Write Cache Policy describes the write cache policy for all physical drives on a controller. The physical device write caching can be enable or disable on all drives on the controller.
Configured Drives	This option allows to configure the write cache policy on a controller. Setting to default allows the controller to optimize the drive write cache policy of those drives. Enabling drive write cache can increase write performance but risks losing the data in the cache on sudden power loss.
Unconfigured Drives	This option allows to configure the write cache policy on a controller. Setting to default for unconfigured drives uses the drive's existing write cache policy. Enabling drive write cache can increase write performance but risks losing the data in the cache on sudden power loss.

Properties	Description
HBA Drives	This option allows to configure the write cache policy on a controller. Setting to default uses the drive's existing write cache policy. Enabling drive write cache can increase write performance but risks losing the data in the cache on sudden power loss.
maxCache Version	maxCache version of controller
maxCache RAID5 WriteBack Enabled	Status of write back setting for RAID5 maxCache at controller level.
Logical devices/Failed/Degraded	Total number of logical devices/number of logical devices failed/number of logical devices failed.
Copyback	When a logical drive is rebuilt using a hot spare, data from the failed drive is transferred to the hot spare. When copyback is enabled, data is moved back to its original location once the controller detects that the failed drive has been replaced. Once the data is copied back, the hot spare becomes available again.
Spare Activation Mode	Spare activation mode feature enables the controller firmware to activate a spare drive. The firmware starts rebuilding a spare drive only when a data drive fails when the mode is Failure. With the predictive failure activation mode, rebuilding can begin before the drive fails when a data drive reports a predictive failure (SMART) status which will reduce the likelihood of data loss that could occur if an additional drive fails.
Background consistency check	If your controller supports background consistency check, maxView Storage Manager continually and automatically checks your logical drives for bad or inconsistent data once they're in use.
Background consistency check period	RAID Consistency Check Period in days
Consistency Check Delay	Determines the time interval for which a controller must be inactive before a consistency check is initiated on the physical drives that are connected to it. The value can be between 0 and 30 to specify the duration of the delay in seconds. A value of 0 disables the scan.
Parallel Consistency Check Supported	Status of the Parallel Consistency Check support
Parallel Consistency Check Count	Parallel consistency check count describes the number of logical devices on which the controller will perform consistency check in parallel.
Inconsistency Repair Policy	Inconsistency repair policy searches for a single inconsistent strip and repairs the strip on that one drive only
Consistency Check Inconsistency Notify	This property enables the event notification messages and serial debug log messages for mirrored volumes
Rebuild Priority	Rebuild priority determines the urgency with which the controller treats an internal command to rebuild a failed logical drive. At the low setting, normal system operations take priority over a rebuild. At the medium setting, rebuilding occurs for half of the time, and normal system operations occur for the rest of the time. At the medium high setting, rebuilding is given a higher priority over normal system operations. At the high setting, the rebuild takes precedence over all other system operations.
Expand Priority	Determines the urgency with which the controller treats an internal command to expand an array. At the low setting level, normal system operations take priority over an array expansion. At the medium setting, expansion occurs for half of the time, and normal system operations occur for the rest of the time. At the high setting, the expansion takes precedence over all other system operations.
Firmware	Active firmware version of the controller
Driver	Current version of driver installed on the system

Properties	Description
Hardware Revision	Describes the hardware revision information about the controller
maxCrypto Supported	maxCrypto feature ensures the sensitive data is encrypted and protected by secure 256 bit AES, in-line encryption
maxCrypto Status	Indicates whether maxCrypto is enabled or disabled
Crypto Officer Password	Crypto officer password indicates, whether the crypto officer(Admin) has configured password or not
User Password	User officer password indicates whether the user has configured password or not
Controller Password	Indicates whether the controller has configured password or not
Allow New Plaintext Logical device(s)	Allow new plaintext logical device(s) indicates, whether the new logical devices created can be encrypted or Plaintext (Non-Encrypted)
Master Key	Master key is a cryptographic secret key
FW Locked for Update	Firmware locked for update prevents the updating of firmware on the controller
Password Recovery Parameters Set	Crypto officer password recovery parameters indicates, whether the crypto officer has set the recovery question and answer
Controller Password Unlock Attempts Remaining	Indicates the number of controller login attempts remaining
SSD I/O Bypass Mixing	maxCrypto/SSD I/O bypass mixing indicates that mixing of maxCrypto and SSD I/O bypass is supported or not
maxCache Mixing	Indicates whether mixing of maxCrypto and maxCache is supported or not
Crypto Account Password Unlock Attempts Remaining	Indicates the number of crypto officer login attempts remaining
User Account Password Unlock Attempts Remaining	Indicates the number of user login attempts remaining.
Number of maxCrypto Physical devices	Total number of encrypted physical device which are part of encrypted logical device
Number of maxCrypto Data Logical devices	Total number of encrypted logical device
Number of maxCrypto Foreign Logical devices without key	Encrypted Foreign Logical Device Count, Total number of encrypted foreign logical device
Number of maxCrypto Logical devices with maxCrypto off	Total number of logical devices without maxCrypto
Firmware Supports Online Firmware Activation	Determines whether firmware supports online firmware activation (OFA) feature
Driver Supports Online Firmware Activation	Determines whether driver supports OFA
OFA Status	Represents the status of OFA
Delay Time Remaining	Determines the delay in minutes for OFA to initiate

Properties	Description
Percentage Complete	Determines the percentage of completion of OFA
Sensor ID	ID describes unique sensor identifier within the controller
Current Value	Current temperature of the specific sensor
Max Value Since Powered On	Maximum temperature of the specific sensor after being powered on the controller
Location	Location of the specific sensor on the controller

**Table 6 • Array**

Properties	Description
Array Number	ID describes unique array identifier within the controller
Status	Status of array is based on health of the member disk drives
Name	Name describes unique name of an array
Transformation Status	Transformation status indicates whether the array is transforming or not
Interface	Disk drives that are the member of array can have interface type such SAS, SATA, SAS S-SD, and SATA SSD. The interface type of array is based on the member disk drives interface type.
Total size	Total usable size is the total space available in the array for creating logical device
Unused Size	Unused size is the free space available to create new logical device to store the data
Block size	Block size indicates the maximum size of data block on disk drives that are member of an array (can be 512 Bytes or 4K)
Array Utilization	Total space utilization of an array
Type	Describes the type of the array such as data array, backup array, and so on
<b>Associated Split Mirror Array</b>	
Spare Rebuild Mode	Spare Rebuild mode describes the spare type for the array. It can be "dedicated" or "auto replace" if the array is valid
SSD I/O Bypass	SSD I/O Bypass enables an optimized data path to high performance solid state drives. The optimized path bypasses the controllers RAID processing components and sends I/O directly to the drives.

**Table 7 • Logical Device**

Properties	Description
Logical Device Number	ID describes unique logical device identifier within the controller.
Logical Device Name	Logical device name can be of maximum 64 characters and it should contain only ASCII characters
Disk Name	Name of the logical disk drive



Properties	Description
Block Size of member drives	Maximum size of data block on disk drives which are RAID member of logical device (can be 512 Bytes or 4K).
Array	Identifier of the Array on which the logical device.
RAID level	RAID level on which the logical device has been created.
Unique Identifier	The logical device unique identifier.
Status of Logical Device	Status of logical device based on health of RAID members of logical device.
maxCache state	State of the associated maxCache logical device.
Additional details	Describes the way of logical drive creation
Parity Initialization Status	RPI status of this logical device
maxCache Status	Status of maxCache.
Size	Logical drive size
Parity Space	the parity space of the logical device
Stripe-unit size	Stripe size is the amount of data (in KB) written to one disk drive, before moving to the next disk drive in the logical device. Stripe size options vary, depending on your controller and RAID level.
Full Stripe Size	Full stripe size refers to the combined size of all the strips across all physical drives, excluding parity-only drives.
Cache Line Size	Specifies the cache line size(KB)
Interface Type	Disk drives which are the member of logical can have interface type such SAS, SATA, SAS SSD and SATA SSD.
Device Type	Indicates logical drive is of type data or cache
Bootable	Specifies logical drive bootable type
Failed stripes	Displays true if bad blocks on logical drive exists
Failed stripe count	Shows total bad blocks on logical drive
Protected by Hot Spare	Indicates whether the logical device is protected by Hot Spare.
Heads	Indicates the pre-defined space set aside for RAID redundant information on a logical device.
Sectors Per Track	Specifies the number of sectors that are to comprise each track
Cylinders	Indicates the set of all tracks having equal diameter in a logical device
Caching	Indicates the setting of cache used for this logical device.
Mount Points	Describes the operating system device names of the logical device
Associated Logical Id	Associated logical device Id for maxCache

Properties	Description
maxCache write cache policy	maxCache policy setting to WT or WB
Backup Creation Timestamp	Split Mirror Set Backup logical creation time
LD Acceleration Method	Setting of the LD acceleration method. Controller cache or SSD I/O Bypass or maxCache
Volume Unique Identifier	Logical device unique string identifier
Plaintext (Non-Encrypted) Logical Device	Setting to identify a non-encrypted logical device
Volatile maxCrypto Keys	Setting to identify if logical device supports Volatile key.
Cache:	maxCache statistics
Read Cache Hits	Read Cache Hits
Read Cache Miss (Total)	Total read cache miss
Read Cache Hit Rate	Read cache hit rate
Write Cache Hits	Write cache hits
Write Cache Miss (Total)	Write cache miss
Write Cache Hit Rate	Write cache hit rate
Power settings	Power setting information (The total board power, which is the total power of all the power sensors plus a static board value.)
Boot Type	A bootable logical device is a logical device that the system can attempt to boot from after a system power-on. A controller can have up to two bootable logical device, where one is a primary boot logical device and the other a secondary boot logical device. When the system looks at a controller for a boot logical device, it will first attempt to boot from a primary boot logical device, and if that fails, then it will attempt to boot from a secondary boot logical device.
maxCrypto	setting of the maxCrypto for the logical device
Encrypted	Indicate user whether the logical device is encrypted or not.

**Table 8 • Physical Device**

Properties	Description
Device	The SCSI ID for a physical device reported by the controller
Device Type	Type of physical device such as "hard disk drive", "solid state drive" or "shingled magnetic recording (SMR) hard disk drive"
State	Current state of the physical device based on the operations done on it
Drive has stale RIS data	Specifies whether the physical device has stale RIS data
Disk name	Name of the physical disk drive
Block Size	Maximum size of the data block on disk drives that are RAID member of the logical device (can be 512 Bytes or 4K)

Properties	Description
Failed logical device segments	If dead chunk in the hard drive found
Dedicated Spare for drive	If drive is acting as spare
Programmed Max Speed	Description of the technology and speed of a device
Transfer Speed	Description of the drive speed
Reported Channel,Device(T:L)	The channel ID and SCSI device ID to which the physical device is connected
Vendor	Physical device manufacturer name
Model	Product model; name of the physical device
Firmware	Firmware version of the physical device
Serial number	Serial number of the physical device
World Wide Name	Reported world wide name provided by the manufacturer
Total Size	Total data storage capacity of the physical device
Reserved Size	Reserved space for internal use
Used Size	Used space of the drive
Unused Size	Unused space of the drive
S.M.A.R.T.	SMART supported
S.M.A.R.T. warnings	Any SMART error reported in the physical device
SSD	If drive type SSD
Device-On-Boot Connector	Device connected through the boot connector
NCQ status	Indicates whether the native command queuing is enabled/disabled on this physical device
NCQ supported	Specifies whether this physical device supports native command queuing
Rotational Speed	Indicates the rotational speed of the physical device
Current Temperature	Current temperature of the physical device
Maximum Temperature	The maximum temperature reported by the physical device
PHY Count	Drive PHY count
Drive Configuration Type	Determines the presence/type of logical devices of which this physical device is a part
Mount Point(s)	Mount point(s) describes the operating system device names of the physical device
Drive Exposed to OS	Indicates whether the physical device is exposed to the operating system
Sanitize Erase Support	Specifies whether the sanitize erase is supported by this physical device

Properties	Description
Sanitize Lock Freeze Support	Specifies whether the sanitize freeze is supported by this physical device
Sanitize Lock Anti-Freeze Support	Specifies whether the sanitize lock anti-freeze is supported by this physical device
Sanitize Lock Setting	Sanitize Lock Setting information
Usage Remaining	SSD life utilization
SSD Smart Trip Wearout	SSD wear out occurred
56 Day Warning Present	SSD have lesser than or equal to 56 days of the calculated usage left
Drive Unique ID	ID to uniquely identify the physical device
Last Failure Reason	Describes previous drive failure reason
Encrypted	A SED (or Self-Encrypting Drive) is a type of hard drive that automatically and continuously encrypts the data on the drive without any user interaction
Volatile maxCrypto Key	Whether this device has volatile keys
PHY	Drive PHY information
Negotiated Physical Link Rate	Indicating the negotiated physical link rate on all the PHYs of this device
Negotiated Logical Link Rate	Indicating the negotiated logical link rate on all the PHYs of this device
Maximum Link Rate	Maximum possible link rate for a PHY
Negotiated Link Rate	PHY negotiated link rate
<b>Runtime Error Counters</b>	
Hardware Error Count	Hardware Error Count
Medium Error Count	Medium Error Count
Parity Error Count	Parity Error Count
Link Failure Count	Link Failure Count
Aborted Command Count	Aborted Command Count
SMART Warning Count	SMART Warning Count
<b>Device Error Counters:</b>	
Aborted Commands	Aborted Commands
Bad Target Errors	Bad Target Errors
ECC Recovered Read Errors	ECC Recovered Read Errors
Failed Read Recovers	Failed Read Recovers
Failed Write Recovers	Failed Write Recovers

Properties	Description
Format Errors	Format Errors
Hardware Errors	Hardware Errors
Hard Read Errors	Hard Read Errors
Hard Write Errors	Hard Write Errors
Hot Plug Count	Hot Plug Count
Media Failures	Media Failures
Not Ready Errors	Not Ready Errors
Other Time Out Errors	Other Time Out Errors
Predictive Failures	Predictive Failures
Retry Recovered Read Errors	Retry Recovered Read Errors
Retry Recovered Write Errors	Retry Recovered Write Errors
SCSI Bus Faults	SCSI Bus Faults
Sectors Reads	Sectors Reads
Sectors Written	Sectors Written
Service Hours	Service Hours
Interface Type	Interface type supported by the physical device
Boot Type	A bootable physical device is a physical device from which the system attempts to boot after a system power-on. A controller can have up to two bootable physical devices, where one is a primary boot physical device and the other is a secondary boot physical device. When the system looks at a controller for a boot physical device, it will first attempt to boot from a primary boot physical device, and if that fails, then it will attempt to boot from a secondary boot physical device.
Encrypted Drive	Indicates whether this physical device is a part of any encrypted logical device

**Microsemi**

2355 W. Chandler Blvd.  
 Chandler, AZ 85224 USA

Within the USA: +1 (480) 792-7200  
 Fax: +1 (480) 792-7277

www.microsemi.com © 2020 Microsemi and its corporate affiliates. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation and its corporate affiliates. All other trademarks and service marks are the property of their respective owners.

Microsemi's product warranty is set forth in Microsemi's Sales Order Terms and Conditions. Information contained in this publication is provided for the sole purpose of designing with and using Microsemi products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is your responsibility to ensure that your application meets with your specifications. THIS INFORMATION IS PROVIDED "AS IS." MICROSEMI MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROSEMI BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE WHATSOEVER RELATED TO THIS INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROSEMI HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROSEMI'S TOTAL LIABILITY ON ALL CLAIMS IN RELATED TO THIS INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, YOU PAID DIRECTLY TO MICROSEMI FOR THIS INFORMATION. Use of Microsemi devices in life support, mission-critical equipment or applications, and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend and indemnify Microsemi from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microsemi intellectual property rights unless otherwise stated.

Microsemi Corporation, a subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), and its corporate affiliates are leading providers of smart, connected and secure embedded control solutions. Their easy-to-use development tools and comprehensive product portfolio enable customers to create optimal designs which reduce risk while lowering total system cost and time to market. These solutions serve more than 120,000 customers across the industrial, automotive, consumer, aerospace and defense, communications and computing markets. Headquartered in Chandler, Arizona, the company offers outstanding technical support along with dependable delivery and quality. Learn more at [www.microsemi.com](http://www.microsemi.com).

ESC-2161616