Microsemi Smart Storage Controllers

User's Guide

ARCCONF Command Line Utility

Released

Revision 4.0: April 2018

Microsemi

Power Matters.™

Revision History

Revision	Revision Date	Details of Change
4.0	April 2018	Revision 4.0 is a post-production release of this document published in March 2018. The following is a summary of the changes: • Add commands: maxcrypto, maxcryptoaccounts, maxcryptokey • arcconf Create-Add ENCODE parameter • arcconf SetControllerParam-Add MIXEDVOLUMES, FWLOCK parameters • arcconf Task-Add ENCODE, CRYPTOERASE, REKEY parameters • arcconf SetConfig-Add CLEARMAXCRYPTOCONFIG parameter • arcconf GetConfig-Display maxCrypto properties • Add section: Running ARCCONF from the UEFI Shell • Add commands supported in UEFI/ARCCONF only: passthrough, slotconfig
3.0	September 2017	Revision 3.0 is a post-production release of this document published in September 2017. The following is a summary of the changes: • arconf getconfig-Added logical drive maxCache statistics to the list of information provided by this command. • arcconf setarrayparam-Added SSDIOBYPASS to parameter. • arcconf setcontrollerparam-Added SANITIZELOCK parameter. • arcconf task-Added CHANNEL# ID# parameter and options for secureerase. • arcconf uart-Removed this command.
2.0	February 2017	Revision 2.0 is a post-production release of this document published in February 2017. The following is a summary of the changes: • Downloading the Installation Packages-Removed ".exe" from the Linux command. • arcconf create-Added maxCache support and added RAID levels. Added note to avoid mixing SMR and PMR in an array. • arcconf getconfig-Added display of controller manufacturing information, green backup, associated split mirror array information, and I2C address, clock speed and clock stretching information. • arcconf identify-Added display of logical drive and array. • arcconf modify-Added note to avoid mixing SMR and PMR in an array. • arcconf romupdate-Updated download URL for .bin file. • arcconf setcontrollerparam- Added I2CADDRESS. • arcconf setmaxcache-Added this new command. • arcconf setpower-Added this new command.
1	August 2016	Preliminary Release for Early Customer Engagement.

Contents

Tables

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

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
- Provides access to the status and event logs of a controller

Note: This guide focuses on using ARCCONF with Microsemi 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 `arcconf` has 'execute' privilege:
   `chmod arcconf +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 arcconf 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).
2. Run arcconf 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 52.

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

Options	Description
Windows	`<install_dir>\arcconf.exe`
Linux	`/<install_dir>/arcconf`

Options	Description
VMware ESXi with Remote ARCCONF	`/usr/RemoteArcconf/arcconf`

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)
```

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 Microsemi 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[2]
delete	maxcrypto[1]	setbiosparams	smp
driverupdate	maxcryptoaccounts[1]	setboot	splitmirror
expanderlist	maxcryptokey[1]	setcache	task
expanderupgrade	modify	setconfig	uninit
getconfig	passthrough[2]	setconnectormode	
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, the firmware returns an error.

[1] Available on controllers that support maxCrypto Controller-Based Encryption. See the Release Notes for more information.

[2] Available in UEFI/ARCCONF only. See Running ARCCONF in the UEFI Shell on page 52.

2.2 arcconf atapassword

Description

Sets or clears the password for SATA drives.

Syntax

```
ARCCONF ATAPASSWORD <Controller#> SET <new password> <Channel# ID#> ...
ARCCONF ATAPASSWORD <Controller#> CLEAR <current password> <Channel# ID#> ...
```

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 arcconf consistencycheck

Description

Toggles the background consistency check modes of the controller.

Syntax

```
ARCCONF CONSISTENCYCHECK <Controller#> <on [Delay]|off> [noprompt]
ARCCONF CONSISTENCYCHECK <Controller#> PARALLELCOUNT <Count>
```

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 surface scan count for the controller. A value of 1 disables the consistency check.

Noprompt

Optional parameter that suppresses the confirmation prompt.

Examples

```
ARCCONF CONSISTENCYCHECK 1 OFF
ARCCONF CONSISTENCYCHECK 1 PARALLELCOUNT 4
```

2.4 arcconf 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#
DRIVE#> [CHANNEL# DRIVE#] ... [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#> [Channel1# ID#]... [noprompt] [nologs]
ARCCONF CREATE <Controller#> MAXCACHE [Options] DATALD, <LogicalDrive#> <Size>
 <RAID#> ARRAY <ARRAY#> [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.
- 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 16 characters.
- Method <METHOD>—Initialization method for the logical drive. Valid options include: BUILD, .
- LDcache—Sets the cache state for the logical drive:
  - LON— cache on
  - LOFF—cache off

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 24):

- 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, with the following options:
- Sectors <sectors>—Sectors per track of the logical device. Valid options are 32 and 63.
- SSDOverProvisioningOptimization <enable | disable>—Initializes 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# Drive#

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 datald 0 17000 1 0 0 0 1
ARCCONF CREATE 1 MAXCACHE datald 0 17000 0 ARRAY 0
ARCCONF CREATE 1 LOGICALDRIVE ENCODE disable USERROLE crypto PASSWORD Abc@1234 MAX 5 0
 0 0 1 0 2
```

2.5 arcconf delete

Description

Deletes a logical drive. All data stored on the logical drive will be lost.

Syntax

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

LogicalDrive ALL
: Deletes all logical drives.

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 arcconf 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.

Nologs
: Optional parameter that suppresses log output.

Examples

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

2.7 arcconf expanderlist

Description

Returns a list of disk drive expanders on a controller.

Syntax

```
ARCCONF EXPANDERLIST <Controller#>
```

Parameters

Controller#
Controller number.

Examples

```
ARCCONF EXPANDERLIST 1
```

2.8 arcconf 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]
```

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 65536 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.

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 arcconf getconfig

Description

Lists information about controllers, physical drives, including:

- 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 segment and group information
- maxCache status, fetch and flush rate policy, read/write balance, 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)
- 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

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]
ARCCONF GETCONFIG <Controller#> [AR [AR#]|CN]
```

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

Channel# ID#

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

Examples

```
ARCCONF GETCONFIG 1
   ---------------------------------------------------------
   Controller BIOS Setting Information
   ---------------------------------------------------------
   Runtime BIOS                              : Enabled
   Array BBS Support                         : Enabled
   Physical Drives Displayed during POST     : Disabled
   Backplane Mode                            : SGPIO
   MissingDrvCount                           : 8
   ...
   Overall Green Backup Unit Status          : Ok
   Battery/Capacitor Pack Count              : 1
   Battery Power Status                      : Charged
   Hardware Error                            : No Error
   Preservataion Status                      : Enabled
   Preservataion Set                         : Disabled
   Version (Major:Minor)                     : (6:0)

GETCONFIG 1
GETCONFIG 1 AD
GETCONFIG 1 LD
GETCONFIG 1 LD 0
GETCONFIG 1 PD
GETCONFIG 1 PD 0 0
GETCONFIG 1 AR
GETCONFIG 1 AR 0
GETCONFIG 1 MC
GETCONFIG 1 CN
GETCONFIG 1 AL
```

2.10 arcconf getlogs

Description

Provides access to controller status, event logs, and usage statistics, including:

- A log of device errors that the controller encountered
- A log that records any occurrences of defunct devices
- A log of special events that may have occurred (rebuilds, LDMs, etc.)
- A log of controller usage statistics, including Inter-I/O Read and Write times and I/O Completion Read and Write times
- A log of cache statistics for one or all logical drives

- A log of supported hardware components on the controller

Syntax

```
ARCCONF GETLOGS <Controller#> <Type1> [clear|tabular] [nologs]
ARCCONF GETLOGS <Controller#> <Type2> [tabular] [nologs]
ARCCONF GETLOGS <Controller#> <Type3> LOGICALDRIVE [<LD#>|ALL] [tabular]
[nologs]
ARCCONF GETLOGS <Controller#> <Type4> <HardwareType#> [nologs]
ARCCONF GETLOGS <Controller#> DEVICE <clear> <ErrorType#> [<Channel# ID#>|ALL]
 [nologs]
```

Parameters

Controller#

Controller number.

Type1

One of the following:

- DEVICE—device error log
- DEAD—dead (failed) drive log
- EVENT—controller event log

Type2

- STATS—controller statistics data

Type3

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

Type4

One of the following:

1. Cache Memory
2. NVSRAM

ErrorType

One of the following:

- 1—Parity Error Counter
- 2—Link Failure Counter
- 3—Hardware Error Counter
- 4—Aborted Commands Counter
- 5—Medium Error Counter
- 6—SMART Warning Counter

Clear

Clears the specified log from the controller or a specific error counter for one or all physical drives on a controller.

Channel/ID

Channel and number of the physical device on the controller.

Tabular

Displays the log or statistics in tabular format.

Examples

```
ARCCONF GETLOGS 1 EVENT
ARCCONF GETLOGS 1 STATS tabular
ARCCONF GETLOGS 1 DEVICE clear 3 ALL
```

2.11 arcconf getsmartstats

Description

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

Syntax

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

Parameters

Controller#
Controller number.

Tabular
Creates output in tabular format.

Examples

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

2.12 arcconf getstatus

Description

The GETSTATUS function displays the status of any background command 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 arcconf getversion

Description

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

Syntax

```
ARCCONF GETVERSION
ARCCONF GETVERSION <Controller#>
```

Parameters

Controller#

Controller# is the controller number

Examples

```
ARCCONF GETVERSION
ARCCONF GETVERSION 1
```

2.14 arcconf identify

Description

Identifies a physical device by blinking its LEDs.

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]
```

Parameters

Controller#

Controller number

LogicalDrive#

Number of the logical drive to be identified

Array#

Array number

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]
```

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-(SATA) Download with offsets and save image for immediate and future use
- 7-(SAS) Download microcode with offsets, save, 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 Microsemi controller.

Syntax

```
ARCCONF KEY <Controller#> SET <Key#>
```

Parameters

Controller#
The controller number.

Key#
The key number provided by Microsemi.

Examples

```
ARCCONF KEY 1 SET ABCDEFGHIJKLMNOPQRSTUVWX
```

2.17 arcconf list

Description

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

Syntax

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

Parameters

Controller#
The controller number.

Examples

```
ARCCONF LIST
ARCCONF LIST 1
```

2.18 arcconf maxcrypto

Description

Configures maxCrypto settings, including:

- maxCrypto master key
- Mode (enable/disable)
- Administrator account credentials
- Support for mixed encrypted/plaintext 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>]
```

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

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 Abc@1234567 USERROLE crypto PASSWORD Abc@123456
ARCCONF MAXCRYPTO 1 SETUP manual MODE disable KEYMANAGEMENTMODE local MIXEDVOLUMES enable
```

```
 MASTERKEY Abc@1234567 USERROLE crypto password Pmc@1234
ARCCONF MAXCRYPTO 1 MODE disable USERROLE crypto PASSWORD Abc@1234
```

2.19 arcconf 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>]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> CHANGEPASSWORD USERROLE <crypto | user>
 [OLDPASSWORD <crypto/user password> NEWPASSWORD <crypto/user password>]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> SETRECOVERYPARAM QUESTION <"Question">
 ANSWER <"Answer"> USERROLE crypto [PASSWORD <crypto password>]
ARCCONF MAXCRYPTOACCOUNTS <Controller#> RECOVERPASSWORD ANSWER <"Answer">
USERROLE crypto [NEWPASSWORD <crypto password>]
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 arcconf setcontrollerparam on page 40.

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 arcconf 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>]
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>]
ARCCONF MAXCRYPTOKEY <Controller#> IMPORT MASTERKEY <masterkey> USERROLE <crypto
 | user> [PASSWORD <crypto/user password>]
```

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.

USERROLE <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 USERROLE crypto PASSWORD Abc@123456
```

```
ARCCONF MAXCRYPTOKEY 1 REKEY ARRAY 0 USERROLE crypto PASSWORD Abc@123456
```

2.21 arcconf 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]
```

```
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]
```

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.

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
```

2.22 arcconf 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

```
Usage:
PASSTHROUGH <Controller#> <Channel# ID#> <read|notransfer> [length] <[> CDB>
<]> [rawhex] [noprompt] [nologs]
PASSTHROUGH <Controller#> <Channel# ID#> <readwithsense|notransferwithsese>
[length] <[> <CDB> <]> [rawhex] [noprompt] [nologs]
PASSTHROUGH <Controller#> <Channel# ID#> <write> [length] <[> <CDB> <]>
<Filename> [rawhex] [noprompt]
PASSTHROUGH <Controller#> <Channel# ID#> <writewithsense> [length] <[> <CDB>
<]> <Filename> [rawhex] [noprompt]
PASSTHROUGH <Controller#> <Channel# ID#1-ID#2> <read> [length] <[> <CDB> <]>
[rawhex] [noprompt]
PASSTHROUGH <Controller#> <Channel# ID#1, ID#2, #ID#3> <read> [length] <[>
<CDB> <]> [rawhex] [noprompt]
```

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.

notransfer
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.

notransferwithsense
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

Table 3 • Passthrough CDB Commands

Opcode	Command	Direction
0x00	TEST UNIT READY	notransfer
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	notransfer
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

Opcode	Command	Direction
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 arcconf phyerrorlog

Description

Displays PHY error logs for physical devices on a controller.

Syntax

```
ARCCONF PHYERRORLOG <Controller#> DEVICE <Channel# ID#>
ARCCONF PHYERRORLOG <Controller#> DEVICE ALL
```

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 arcconf playconfig

Description

Configures a controller using a XML server template file produced by the SAVECONFIG command (see arcconf saveconfig on page 34). 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.
SUCCESS_REBOOT	2	Configuration succeeded, but a reboot is required.

Syntax

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

Parameters

Input XML File

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

LogFile

Sets the pathname of the error log file. By default, the error log is available at C:\PMCS\Logs\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 arcconf 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 arcconf 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#>
```

Parameters

Controller#
The controller number

Examples

```
ARCCONF RESETSTATISTICSCOUNTERS 1
```

2.27 arcconf 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.
2. You can download the .bin update files at www.start.microsemi.com

Syntax

```
ARCCONF ROMUPDATE <Controller#> <BaseName> [newversion <build#> [force]]
[noprompt] [nologs]
```

Parameters

Controller#
The controller number.

BaseName
Absolute path to the controller image file.

Newversion <build#>
Specifies the version of the firmware build.

Force
An optional parameter used to force a down-level firmware update. Valid only if Newversion parameter is specified.

Noprompt
An optional parameter that suppresses the confirmation prompt.

Examples

```
ARCCONF ROMUPDATE 1 /usr/home/AC220001.BIN
ARCCONF ROMUPDATE 1 C:\firmwareImage\as483c.bin newversion 12345 force noprompt
```

2.28 arcconf 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 arcconf playconfig on page 32 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]
```

Parameters

Input XML File

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

LogFile

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

Examples

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

2.29 arcconf 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 SSD 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|Arcconf|Storlib|Basecode]
```

Parameters

Path

Path to store the log files.

Log type:

One of these log files:

- Firmware: saves Firmware logs
- Arcconf: saves Arcconf logs
- Storlib: saves StorLib logs
- Basecode: saves basecode logs

Examples

```
ARCCONF SAVESUPPORTARCHIVE
ARCCONF SAVESUPPORTARCHIVE Firmware
```

2.30 arcconf setarrayparam

Description

Changes a parameter of an array.

Syntax

```
ARCCONF SETARRAYPARAM <Controller#> <Array#> SPARETYPE <Type> [nologs]
ARCCONF SETARRAYPARAM <Controller#> <Array#> CONSOLIDATESPACE [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 dedicated 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.

CONSOLIDATESPACE

Relocates the logical drives in the array and consolidates the array free space at the end of the array.

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 arcconf setbiosparams

Description

Changes select BIOS settings

Syntax

```
ARCCONF SETBIOSPARAMS <Controller#> POSTPROMPTTIMEOUT <timeout>
```

Parameters

Controller#

Controller number

Subfunction

Option	Description
POSTPROMPTTIMEOUT <timeout>	Changes the F1/F2 POST prompt timeout for the controller during system boot. Timeout can have a value between 1-255.

Examples

```
ARCCONF SETBIOSPARAMS 1 POSTPROMPTTIMECOUNT 10
```

2.32 arcconf 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
```

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
```

2.33 arcconf 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#> DEVICEALL <policy> [noprompt] [nologs]
ARCCONF SETCACHE <Controller#> CACHERATIO <read#> <write#>
ARCCONF SETCACHE <Controller#> WAITFORCACHEROOM <enable | disable>
ARCCONF SETCACHE <Controller#> NOBATTERYWRITECACHE <enable | disable>
```

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.

Policy

- Enable - write back for all physical drives
- Disable - write through for all physical drives

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.

Examples

```
ARCCONF SETCACHE 1 DEVICEALL Enable
ARCCONF SETCACHE 1 CACHERATIO 60 40
ARCCONF SETCACHE 1 WAITFORCACHEROOM enable
ARCCONF SETCACHE 1 NOBATTERYWRITECACHE enable
```

2.34 arcconf setconfig

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 maxCryto configuration, including all keys, passwords, and maxCrypto users (administrator and standard user).

Syntax

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

Parameters

Controller#
: The controller number

Default
: Restores the controller's default configuration.

Clearmaxcryptoconfig
: Restores the default maxCryto configuration.

Noprompt
: No prompt for confirmation.

Examples

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

2.35 arcconf setconnectormode

Description

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

- 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.
- Smart HBA (Mixed)—RAID volumes and RAW drives 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 - Smart HBA (Mixed)

nologs

Suppresses log output.

Examples

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

2.36 arcconf 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#> I2CADDRESS <i2cAddress> <i2cClockSpeed>
 <i2cClockStretching> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> SANITIZELOCK <sanitizeLock>
ARCCONF SETCONTROLLERPARAM <Controller#> MIXEDVOLUMES <Enable | Disable>
USERROLE <userrole> PASSWORD <password>
ARCCONF SETCONTROLLERPARAM <Controller#> FWLOCK <Enable | Disable> USERROLE
<userrole> PASSWORD <password>
```

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.

I2CADDRESS

Sets the I2C Address of the controller. Sets the I2C clock speed.

- i2cAddress. Hexadecimal input from range of 0x00–0xFF.
- i2cClockSpeed. Sets the I2C clock speed
  - 2 : I2C Clock Speed 100 kHz
  - 3 : I2C Clock Speed 400 kHz
- i2cClockStretching. Sets the I2C clock stretch.
  - Enable—Enables clock stretching.
  - Disable—Disables clock stretching.

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)

Examples

```
ARCCONF SETCONTROLLERPARAM 1 QUEUEDEPTH 16
ARCCONF SETCONTROLLERPARAM 1 SPAREACTIVATIONMODE 0
ARCCONF SETCONTROLLERPARAM 1 ELAVATORSORT disable
ARCCONF SETCONTROLLERPARAM 1 LATENCY 2
ARCCONF SETCONTROLLERPARAM 1 I2CADDRESS 0x05 2 Disable
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
```

2.37 arcconf setmaxcache

Description

Enables/disables maxCache SSD caching for one or more logical drives; updates the maxCache write cache policy and “dirty page” threshold (data not committed to disk); adds Solid State Drives to the maxCache pool and removes SSDs from the pool; sets the maxCache read/write balance and cache fetch/flush rate; clears the maxCache pool.

> **Note:** Before you can enable maxCache SSD caching, you must assign at least one SSD to the maxCache pool.

Syntax: Read Caching

```
ARCCONF SETMAXCACHE <Controller#> LOGICALDRIVE <LogicalDrive#> READCACHE
<ENABLE|DISABLE>
```

Syntax: Write Caching

```
ARCCONF SETMAXCACHE <Controller#> LOGICALDRIVE <LogicalDrive#> WRITECACHE
<ENABLE|DISABLE> [WRITEPOLICY <policy>]
ARCCONF SETMAXCACHE <Controller#> LOGICALDRIVE ALL WRITECACHE DISABLE
[WRITEPOLICY <policy>]
ARCCONF SETMAXCACHE <Controller#> LOGICALDRIVE <LogicalDrive#> WRITEPOLICY
<policy>
ARCCONF SETMAXCACHE <Controller#> DIRTYPAGETHRESHOLD <dirtyPageThreshold#>
ARCCONF SETMAXCACHE <Controller#> WBCVALID <ENABLE|DISABLE>
```

General Usage

```
ARCCONF SETMAXCACHE <Controller#> <ADDTOPOOL|REMOVEFROMPOOL> <Channel# Device#>
ARCCONF SETMAXCACHE <Controller#> RWBALANCE <Read#> <Write#>
ARCCONF SETMAXCACHE <Controller#> FLUSHANDFETCHRATE <FlushAndFetchRate#>
ARCCONF SETMAXCACHE <Controller#> CLEAR
```

Parameters

Controller#

The controller number.

LogicalDrive#

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

Channel#

The channel number for the SSD.

Device#

The device number for the SSD.

Read#/Write#

The read/write ratio for invalidating data on the SSD. When the ratio is reached, the page is removed from the cache. Values range from 1-10 for each parameter.

FlushAndFetchRate#

The read cache fetch rate from 1 to 1000: 1-50=Low, 51-100=Medium, 101-1000=High. The default is 100.

Note: The lower the rate the longer the page is kept on the SSD before it is flushed from the cache.

dirtyPageThreshold#

Controls the amount cache space allocated to “dirty” data; that is, data that has not been committed to disk. The threshold value ranges from 1-100 (percent). Once the percentage of dirty pages crosses the threshold, the data are flushed to disk.

WBCVALID ENABLE|DISABLE

Enables and disables write caching in non-redundant maxCache. Applies to all logical drives.

Policy

maxCache write cache policy:

- WB - write back enabled. maxCache will store the data on the SSD and write it back to the hard disks when there is little or no impact on performance. This is the default policy.
- INSTWB - instant write back enabled. In addition to the default policy, maxCache will create dirty pages on-the-fly for full-stripe writes if there is room on the SSD and the number of dirty pages is below the threshold.
- WT - write through enabled. Similar to instant write back, but full-stripe writes go to both the cache and hard disk and no dirty pages are created on-the-fly.

Clear

Clears the maxCache pool.

Examples

```
ARCCONF SETMAXCACHE 1 LOGICALDRIVE 1 READCACHE ENABLE
ARCCONF SETMAXCACHE 1 LOGICALDRIVE 1 WRITECACHE ENABLE WRITEPOLICY WT
ARCCONF SETMAXCACHE 1 DIRTYPAGETHRESHOLD 50
ARCCONF SETMAXCACHE 1 ADDTOPOOL 0 1
ARCCONF SETMAXCACHE 1 REMOVEFROMPOOL 0 1 0 2
ARCCONF SETMAXCACHE 1 RWBALANCE 4 1
ARCCONF SETMAXCACHE 1 FLUSHANDFETCHRATE 200
ARCCONF SETMAXCACHE 1 CLEAR
```

2.38 arcconf setname

Description

Renames a logical drive.

Syntax

```
ARCCONF SETNAME <Controller#> LOGICALDRIVE <LogicalDrive#> <New Name>
```

Parameters

Controller#

Controller number

LogicalDrive#

The number of the logical drive to be renamed

New Name

The new name of the logical drive

Examples

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

2.39 arcconf setperform

Description

Changes controller settings based on the application type.

Syntax

```
ARCCONF SETPERFORM <Controller#> MNPDELAY <Delay> [no logs]
ARCCONF SETPERFORM <Controller#> DPO <Enable | Disable> [no logs]
```

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 arcconf setpower

Description

Modifies the power management settings.

Syntax

```
ARCCONF SETPOWER <Controller#> POWERMODE <mode> SURVIVALMODE <mode>
```

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

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

2.41 arcconf setpriority

Description

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

Syntax

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

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 arcconf setstate

Description

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

Syntax

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

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.

ADVANCED <option>

Optional keyword/option pair. Set option to Nocheck to force a logical drive to the Optimal state without performing a consistency check.

Caution: Using Advanced options may result in data loss!

Noprompt:

No prompt for confirmation.

Examples

```
ARCCONF SETSTATE 1 DEVICE 0 0 RDY LOGICALDRIVE 2
ARCCONF SETSTATE 1 LOGICALDRIVE 1 OPTIMAL ADVANCED nocheck
ARCCONF SETSTATE 1 DEVICE 0 0 DDD
ARCCONF SETSTATE 1 DEVICE 0 0 RDY
ARCCONF SETSTATE 1 DEVICE 0 0 HSP ARRAY 0
```

2.43 arcconf setstatsdatacollection

Description

Enables or disables statistics collection for a controller. To display the statistics, seearcconf getlogs on page 19 .

Syntax

```
ARCCONF SETSTATSDATACOLLECTION <Controller#> Enable|Disable
```

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

```
Usage: SLOTCONFIG <Controller#> <EnclosureID#> <Slot#>
Usage: SLOTCONFIG <Controller#> <EnclosureID#> MAP
```

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.

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

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 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 arcconf 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
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> REMIRROR
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> ROLLBACK
ARCCONF SPLITMIRROR <Controller#> ARRAY <Array#> ACTIVATEBACKUP
```

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 arcconf task

Description

Performs a task on a physical drive. 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
TASK START <Controller#> DEVICE <Channel# ID#> <options> [unrestricted]
[noprompt] [nologs]
TASK START <Controller#> DEVICE ALL UNINITIALIZE
TASK STOP  <Controller#> DEVICE <Channel#> <ID#>
```

Syntax: maxCrypto Usage

```
ARCCONF TASK
TASK START <Controller#> LOGICALDRIVE <LogicalDrive#> CRYPTOERASE USERROLE
<userrole> PASSWORD <password>
TASK START <Controller#> ARRAY <Array#> CRYPTOERASE USERROLE <userrole> PASSWORD
 <password>
TASK START <Controller#> LOGICALDRIVE <LogicalDrive#> ENCODE DATA <Preserve |
 Discard> USERROLE <userrole> PASSWORD <password>
TASK START <Controller#> LOGICALDRIVE <LogicalDrive#> REKEY USERROLE <userrole>
 PASSWORD <password>
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 - Initializes block to random value then zero.
    - 3: Random Random Zero - Initializes block to random value, next block to random value, then zero.
    - 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 Microsemi 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 arcconf uninit

Description

Uninitializes one or more physical drives. The uninitialize command clears Microsemi 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 52. Additionally, some commands are supported in UEFI/ARCCONF only.

Prerequisites

To run UEFI/ACCONF, 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:

   `arcconf <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 arcconf consistencycheck on page 13	
CREATE	Creates a new logical drive; optionally, enables logical drive read caching, write caching.	see arcconf create on page 14
DELETE	Deletes a logical drive.	see arcconf delete on page 15
EXPANDERLIST	Returns list of disk drive expanders on a controller.	see arcconf expanderlist on page 17

Command	Description	Usage
EXPANDERUPGRADE	Allows new firmware to be flashed to an enclosure or expander.	see arcconf expanderupgrade on page 17
GETCONFIG	Lists information about controllers and physical drives.	see arcconf getconfig on page 18
GETLOGS	Provides access to controller status, event logs, and usage statistics. Usage: GETLOGS <Controller#> <Type 1> Usage: GETLOGS <Controller> <Type 2> LOGICALDRIVE <LD#> see arcconf getlogs on page 19	
GETVERSION	Lists version information for all controllers or a specific controller's software components.	see arcconf getversion on page 21
IDENTIFY	Identifies a physical device by blinking its LEDs.	see arcconf identify on page 22
IMAGEUPDATE	Allows new firmware to be flashed to the hard drive.	see arcconf imageupdate on page 23
LIST	Lists controllers in the system.	see arcconf list on page 24
PASSTHROUGH	Sends SCSI passthrough protocol (SCSI) function request to a disk drive. **Note:** This command is supported in UEFI/ARCCONF only.	see arcconf passthrough on page 29
RESCAN	Check for removal of disk drives and for connection of new disk drives to the controller.	see arcconf rescan on page 33
ROMUPDATE	Allows new firmware and BIOS to be flashed to the controller. Usage: ROMUPDATE <Controller#> <ImagePath> [noprompt] [nologs] see arcconf romupdate on page 34	
SAVESUPPORTARCHIVE	Saves configuration and status information. Usage: SAVESUPPORTARCHIVE [Path] see arcconf savesupportarchive on page 35	
SETARRAYPARAM	Changes a parameter of an array.	see arcconf setarrayparam on page 36
SETBOOT	Sets controller as a boot device for the system.	see arcconf setboot on page 37
SETCONFIG	Resets the controller configuration. Usage: SETCONFIG <Controller#> <DEFAULT> [noprompt] [nologs] see arcconf setconfig on page 39	
SETCONNECTORMODE	Change the connector operating mode.	see arcconf setconnectormode on page 39
SETSTATE	Changes the state of a physical device or logical device.	see arcconf setstate on page 45
SLOTCONFIG	Lists the channel ID and device ID of the devices in each enclosure slot. . A slot with no device is marked as EMPTY. **Note:** This command is supported in UEFI/ARCCONF only.	see arcconf slotconfig on page 47
SMP	Sends a SAS Management Protocol (SMP) function request to a SMP target device.	see arcconf smp on page 47

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

ESC-2161616