

# ARCCONF Command Line Utility User Guide for Adaptec® SmartRAID 4300 Series NVMe RAID Accelerator Controller



# Table of Contents

1. Introduction.....	4
2. Terminology Used in this Guide.....	5
3. Getting Started with the Command Line Utility.....	6
3.1. Installing the Command Line Utility.....	6
3.2. Starting the Command Line Utility.....	6
4. Using the Command Line Utility.....	7
4.1. ARCCONF Commands.....	7
4.2. arccnf consistencycheck.....	8
4.3. arccnf create.....	8
4.4. arccnf delete.....	10
4.5. arccnf getconfig.....	11
4.6. arccnf getconfigjson.....	12
4.7. arccnf getlogs.....	13
4.8. arccnf getsmartstats.....	14
4.9. arccnf getstatus.....	14
4.10. arccnf getversion.....	15
4.11. arccnf identify.....	15
4.12. arccnf imageupdate.....	16
4.13. arccnf list.....	17
4.14. arccnf modify.....	17
4.15. arccnf nvmedriveconfig.....	18
4.16. arccnf passthrough.....	18
4.17. arccnf playconfig.....	21
4.18. arccnf rescan.....	21
4.19. arccnf romupdate.....	22
4.20. arccnf saveconfig.....	23
4.21. arccnf savesupportarchive.....	23
4.22. arccnf security.....	24
4.23. arccnf setarrayparam.....	25
4.24. arccnf setcache.....	25
4.25. arccnf setconfig.....	26
4.26. arccnf setcontrollerparam.....	26
4.27. arccnf setname.....	29
4.28. arccnf setpriority.....	29
4.29. arccnf setstate.....	30
4.30. arccnf splitmirror.....	31
4.31. arccnf task.....	32
4.32. arccnf uninit.....	33
5. Running ARCCONF in the UEFI Shell.....	34
6. Controller Device Status.....	37
7. Controller, Array, and Physical Device.....	39
8. Revision History.....	46

Microchip Information.....	47
Trademarks.....	47
Legal Notice.....	47
Microchip Devices Code Protection Feature.....	47

## 1. Introduction

This guide focuses on using ARCCONF for configuring and managing the NVMe RAID Accelerator.



**Important:** All commands in this document may not be supported for your controller. Check the ARCCONF command Help section for controller specific supported commands.

---

## 2. Terminology Used in this Guide

Because this guide provides information that can be used to manage multiple Microchip NVMe RAID Accelerator Controllers in a variety of configurations, the generic term “storage space” is used to refer to the controller(s), disk drives, and systems being managed with maxView Storage Manager.

For efficiency, the term “component” or “components” is used when referring generically to the physical and virtual parts of your storage space, such as systems, disk drives, controllers, and logical drives.

Many of the terms and concepts referred to in this guide are known to computer users by multiple names. In this guide, this terminology is used:

- Controller (also known as adapter, board, or I/O card)
- Disk drive (also known as hard disk, hard drive, or hard disk drive)
- Non-Volatile Memory Express (also known as NVMe SSDs or Solid-State Drives (SSDs))
- Logical drive (also known as a logical device array)
- Array (also known as a storage pool or container)
- System (also known as a server, workstation, or computer)

### 3. Getting Started with the Command Line Utility

This guide explains how your Microchip NVMe RAID Accelerator controller supports the use of the ARCCONF command line utility.

This utility allows you to:

- Create and delete logical drives
- 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 for configuring and managing the NVMe RAID Accelerator.

#### 3.1 Installing the Command Line Utility

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

##### 3.1.1 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.adaptec.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` or `/usr`, for instance).
5. On Linux systems, ensure that `arccnf` has 'execute' privilege:  

```
chmod +x arccnf
```

#### 3.2 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](#).

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

Option	Description
Windows®	<install_dir>\arccnf.exe
Linux®	/usr/<install_dir>/arccnf

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`.

## 4. 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 4-1.** 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 view the online help for a specific command, type `ARCCONF <command>`, then press Enter.

### 4.1 ARCCONF Commands

The following commands are available in ARCCONF for NVMe RAID Accelerator 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.



**Attention:** ARCCONF supports commands for other controllers that are not listed in the following table. In addition, not all commands in the following table are supported by all NVMe RAID Accelerator controllers. If you attempt to execute any command not listed in the following table, or any unsupported command for your controller, the firmware returns an error.

**Table 4-2.** ARCCONF Commands for NVMe RAID Accelerator Controllers

consistencycheck	imageupdate	saveconfig	setstate
create	list	savesupportarchive	splitmirror
delete	modify	security	task
getconfig	nvmedriveconfig	setarrayparam	uninit
getconfigjson	passthrough	setcache	
getlogs	playconfig	setconfig	
getsmartstats	rescan	setcontrollerparam	
getstatus	romupdate	setname	
getversion		setpriority	
identify			

## 4.2 **arccnf consistencycheck**

### Description

Modifies the background consistency check modes and its parameters of the controller.

### Syntax

```
ARCCNF CONSISTENCYCHECK <Controller#> <On [Delay] | Off> [nologs]
ARCCNF CONSISTENCYCHECK <Controller#> PARALLELCOUNT <Count> [nologs]
ARCCNF CONSISTENCYCHECK <Controller#> EVENTNOTIFY <Enable | Disable> [nologs]
```

### Parameters

#### Controller#

The controller number.

#### On [Delay]

Turns background consistency check on, with optional 0 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.

When set to High, the check runs in parallel to host I/O and might have an impact on performance. When set to idle, the check only runs during periods of host inactivity and does not impact performance.

#### Off

Turns background consistency check off. Consistency check mode is set to Disabled.

#### PARALLELCOUNT <Count>

Sets the parallel consistency check count for the controller. Valid range for parallel consistency check count is 1–16. A count of 1 disables the parallel consistency check. Logical devices are only scanned one at a time.

#### EVENTNOTIFY

Sets the inconsistency event notification and serial debug message generation setting for mirrored logical devices. Valid options: Enable and Disable.

#### noprompt

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

```
ARCCNF CONSISTENCYCHECK 1 OFF
ARCCNF CONSISTENCYCHECK 1 PARALLELCOUNT 4
ARCCNF CONSISTENCYCHECK 1 EVENTNOTIFY enable
ARCCNF CONSISTENCYCHECK 1 ON 2
```

## 4.3 **arccnf create**

### Description

Creates a new logical drive. Refer to the optional parameters to configure the logical device optimally based on your requirements. If the optional parameters are not provided, they are set to default values.



## 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#> RAIDZEROARRAY <Channel# ID#> [Channel# ID#] ... [noprompt] [nologs]
ARCCONF CREATE <Controller#> RAIDZEROARRAY ALL [noprompt] [nologs]
```

## Parameters

### Controller#

The controller number.

### Logical Drive

Indicates a logical drive with the following options:

- **Stripesize <STRIPE>**—Allows the logical device stripe size to be built. Optional parameters for specifying a stripe size. STRIPE is specified in kilobytes. Valid options: 16, 32, 64, and 128. The default is 128 KB.
- **Legs <LEG>**—Optional parameters for specifying number of legs. Value is an integer.
  - **LEG**—Number of legs for RAID level 50.
  - **Default**—2 legs
- **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 device. Valid options: BUILD and DEFAULT.
- **DEFAULT**—Initializes parity blocks in the background while the logical device is available for access by the operating system.
- **BUILD**—Initializes both the data and parity blocks. The logical device remains invisible and unavailable to the operating system until the parity initialization process completes.

### Array#

The array ID on which the logical device needs to be created.

### 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.

### ALL

Create arrays with a RAID0 logical device from the list of configurable physical device(s) connected to the controller.

### SSDOverProvisioningOptimization <SSDOverProvisioningOptimization>

Specifies to initialize SSDs that support rapid parity initialization feature. Valid options: Enable and Disable.

### Size

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

### RAID

RAID level for the new logical device. Valid options are: 0, 1, 10, 5, and 50 are supported. Some controller does not support all RAID levels.

**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 device.

#### noprompt

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

#### Note:

1. Default parameter in most circumstances.

#### 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 RAIDZEROARRAY 0 0 0 1 0 2
ARCCONF CREATE 1 RAIDZEROARRAY ALL
```

## 4.4 arccconf delete

### Description

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

### Syntax

```
ARCCONF DELETE <Controller#> LOGICALDRIVE <LogicalDrive#> [noprompt] [nologs]
ARCCONF DELETE <Controller#> LOGICALDRIVE ALL [noprompt] [nologs]
ARCCONF DELETE <Controller#> ARRAY <Array#> [noprompt] [nologs]
ARCCONF DELETE <Controller#> ARRAY ALL [noprompt] [nologs]
```

### Parameters

#### Controller#

The controller number.

#### LogicalDrive#

Logical device ID.

#### Array#

Array ID.

#### ALL

Deletes all the logical devices and the arrays based on the subcommand.

#### noprompt

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.5 **arcconf getconfig**

### Description

Prints the controller configuration:

- Array status, size and member drives
- Controller type, status, World Wide Name (WWN), and mode
- Device driver and CPLD
- Accelerator ID and Accelerator Firmware
- Logical drive status, RAID level and size
- Logical drive mount points, Consistency Check Information
- RAID 10, 50 segment and group information
- NVMe SSD information
- Device type, Device ID, presence of PFA
- Physical device state, mount point or OS partition for the physical drive will not be there as the controller supports only RAID mode.
- Information of Unconfigured system NVMe drives

**Notes:** 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.

### Syntax

```
ARCCONF GETCONFIG <Controller#> [AD | LD [LogicalDrive#] | AR [Array#] | PD [Channel# ID#
Channel# ID#...] | AL | SD] [nologs]
ARCCONF GETCONFIG <Controller#> [ RAID | SPDM ] [nologs]
```

### Parameters

#### Controller#

The controller number.

#### AL

Displays all the controller configuration. This is the default option.

#### AD

Displays the controller configuration only.

#### LD

Displays the logical device configuration only.

#### LogicalDrive#

Displays configuration of the specified logical device.

#### AR

Displays the array configuration only.

#### Array#

Displays configuration of the specified array.

#### PD

Displays the physical device configuration only.

#### Channel# ID#

The Channel and ID of the physical device to be displayed.

#### **SD**

Displays the system NVMe device information only.

#### **RAID**

Displays the RAID properties.

#### **SPDM**

Displays the SPDM properties.

#### **OOB**

Displays the Out-of-Band (OOB) interface properties.

#### **nologs**

An optional parameter that suppresses log output to the log files.

### **Examples**

```
ARCCONF GETCONFIG 1
ARCCONF GETCONFIG 1 AD
ARCCONF GETCONFIG 1 LD
ARCCONF GETCONFIG 1 LD 0
ARCCONF GETCONFIG 1 PD
ARCCONF GETCONFIG 1 PD 0 0
ARCCONF GETCONFIG 1 AR
ARCCONF GETCONFIG 1 AR 0
ARCCONF GETCONFIG 1 AL
ARCCONF GETCONFIG 1 SD
ARCCONF GETCONFIG 1 RAID
```

## **4.6 arccnf getconfigjson**

### **Description**

Displays the current system configuration in JSON format.

### **Syntax:**

```
ARCCONF GETCONFIGJSON ALL [FILENAME] [nologs]
ARCCONF GETCONFIGJSON <Controller#> [FILENAME] [nologs]
ARCCONF GETCONFIGJSON <Controller#> LOGICALDRIVE <LogicalDrive#> [FILENAME] [nologs]
ARCCONF GETCONFIGJSON <Controller#> DEVICE <Channel# ID#> [FILENAME] [nologs]
ARCCONF GETCONFIGJSON <Controller#> ARRAY <Array#> [FILENAME] [nologs]
ARCCONF GETCONFIGJSON SCHEMA [SCHEMA FILENAME] [nologs]
```

### **Parameters**

#### **Controller#**

Controller number

#### **Channel# ID#**

The channel and ID of the physical device.

#### **LogicalDrive#**

The logical device for which configuration is needed.

#### **Array#**

The array for which configuration is needed.

#### **FILENAME**

Optional parameter that, if specified, outputs the configuration to a JSON file. The provided filename must be accompanied by a .json extension.

#### **ALL**

Displays the configuration in JSON format for all controllers.

**SCHEMA**

Schema about a connected controller configuration and its associated devices.

**SCHEMA FILENAME**

Optional parameter that, if specified, outputs schema to a JSON file. The provided filename must be accompanied by a .json extension.

**nologs**

An optional parameter that suppresses log output to the log files.

**Examples**

```
ARCCONF GETCONFIGJSON ALL
ARCCONF GETCONFIGJSON 1
ARCCONF GETCONFIGJSON 1 LOGICALDRIVE 0
ARCCONF GETCONFIGJSON 1 DEVICE 2 0
ARCCONF GETCONFIGJSON 1 ARRAY 0
ARCCONF GETCONFIGJSON ALL C:\configuration.json
ARCCONF GETCONFIGJSON ALL /var/log/configuration.json
ARCCONF GETCONFIGJSON SCHEMA
ARCCONF GETCONFIGJSON SCHEMA C:\schema.json
ARCCONF GETCONFIGJSON SCHEMA /var/log/schema.json
```

**4.7 arccnf getlogs****Description**

Provides access to event logs including:

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

**Syntax**

```
ARCCONF GETLOGS <Controller#> EVENT[tabular] [nologs]
ARCCONF GETLOGS <Controller#> DEVICE <clear> <Channel# ID#> [nologs]
ARCCONF GETLOGS <Controller#> DEVICE <clear> ALL [nologs]
```

**Parameters****Controller#**

Controller number.

**EVENT**

Controller event log.

**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#**

The Channel and ID of the physical device.

**Tabular**

Displays events in tabular (vs XML) format.

**ALL**

All devices connected to controller.

**nologs**

An optional parameter that suppresses log output to the log files.

## Examples

```
ARCCONF GETLOGS 1 DEVICE clear 0 0
ARCCONF GETLOGS 1 DEVICE clear ALL
ARCCONF GETLOGS 1 EVENT tabular
```

## 4.8 arccconf getsmartstats

### Description

Obtains SMART statistics of all the hard drives attached to the controller.

### Syntax

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

### Parameters

#### Controller#

Controller number.

#### Tabular

Displays the specified log in tabular (vs XML) format.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.9 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#> [json] [nologs]
ARCCONF GETSTATUS <Controller#> LOGICALDRIVE ALL [nologs]
ARCCONF GETSTATUS <Controller#> LOGICALDRIVE <LogicalDrive#> [LogicalDrive#] ... [nologs]
ARCCONF GETSTATUS <Controller#> DEVICE ALL [nologs]
ARCCONF GETSTATUS <Controller#> DEVICE <Channel# ID#> [Channel# ID#] ... [nologs]
```

### Parameters

#### Controller#

Controller number.

#### Channel# ID#

A list of space delimited channel number and device number pairs for each device to get the currently running task(s) on the physical device.

#### LogicalDrive#

A list of space-delimited logical drive number to get the currently running task(s) on the logical device.

#### ALL

Displays task information for all devices.

**json**

Displays the status of any background task that is currently running in JSON format.

**nologs**

An optional parameter that suppresses log output to the log files.

**Examples**

```
ARCCONF GETSTATUS 1 JSON
ARCCONF GETSTATUS 1
ARCCONF GETSTATUS 1 LOGICALDRIVE ALL
ARCCONF GETSTATUS 1 LOGICALDRIVE 0 1
ARCCONF GETSTATUS 1 DEVICE ALL
ARCCONF GETSTATUS 1 DEVICE 0 0 0 1
```

## 4.10 arccnf getversion

**Description**

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

**Syntax**

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

**Parameters****Controller#**

Controller number

**Examples**

```
ARCCONF GETVERSION
ARCCONF GETVERSION 1
```

## 4.11 arccnf identify

**Description**

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

**Syntax**

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

**Parameters****Controller#**

Controller number

**LogicalDrive#**

Device number of the logical drive to be identified

**Array#**

Array number to be identified

**Channel# ID#**

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

**ALL**

Blinks all physical devices connected to the controller

**TIME <BlinkTime>**

Time, in seconds, for the LEDs to continue blinking. Maximum time limit is 86400 seconds.

**START**

Starts blinking the device until the STOP command is issued

**STOP**

Stops blinking the device(s)

**nologs**

An optional parameter that suppresses log output to the log files.

**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
```

## 4.12 arccnf imageupdate

**Description**

Allows new firmware to be flashed to the hard drive.

**Syntax:**

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

**Parameters****Controller#**

Controller number.

**Channel# ID#**

The Channel and ID of the physical device whose firmware will be updated.

**ChunkSize#**

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

**Mode#**

Firmware update mode. Valid option: 239(E+F)

- 239(E+F)—Download microcode in 'ChunkSize' byte chunks and activate

**Filename**

Absolute path of the device firmware update file.

**noprompt**

An optional parameter that suppresses the user confirmation prompt.

**nologs**

An optional parameter that suppresses log output to the log files.



**Examples**

```
ARCCONF IMAGEUPDATE 1 DEVICE 0 0 32768 C:\drive_firmware.bin 239
```

**4.13 arccnf 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
```

**4.14 arccnf modify****Description**

Expands a logical device FROM original TO one with larger capacity (Online Capacity Expansion).  
Expands an array.

**Syntax**

```
ARCCONF MODIFY <Controller#> FROM <LogicalDrive#> TO [Options] <Size> <RAID#> <Channel# ID#>
[Channel# ID#] ... [noprompt] [nologs]
ARCCONF MODIFY <Controller#> ARRAY <Array#> EXPAND [modifyparitygroups] <Channel# ID#>
[Channel# ID#] ... [nologs]
```

**Parameters****Controller#**

The controller number

**Size**

- Size of the logical drives in MB.
- Use MAX to set size to all available space.

**RAID#**

RAID level for Expansion logical device.

**ARRAY <Array#>**

The arrayID of the array to be modified.

**EXPAND**

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

**modifyparitygroups**

This option specifies reconfiguration of parity groups on the logical devices based on the final number of physical device(s) in the array.

Parity group modification will not be performed if this option is not specified.

**noprompt**

An optional parameter that suppresses the user confirmation prompt.

#### **nologs**

An optional parameter that suppresses log output to the log files.

#### **Examples**

```
ARCCONF MODIFY 1 FROM 0 TO 1024 0 0 0 0 1
ARCCONF MODIFY 1 ARRAY 1 EXPAND 0 0 0 1
ARCCONF MODIFY 1 ARRAY 1 EXPAND MODIFYPARITYGROUPS 0 0 0 1
```

## **4.15 arccnf nvmedriveconfig**

### **Description**

Manages the NVMe drives for NVMe RAID Accelerator.

### **Syntax**

```
ARCCONF NVMEDRIVECONFIG <Controller#> ADD <Channel# ID#> <LABEL LABEL#> [Channel# ID# <LABEL LABEL#>] ... [noprompt] [nologs]
ARCCONF NVMEDRIVECONFIG <Controller#> REMOVE <Channel# ID#> [Channel# ID#] ... [noprompt] [nologs]
ARCCONF NVMEDRIVECONFIG <Controller#> REMOVE ALL [noprompt] [nologs]
```

### **Parameters**

#### **Controller#**

The controller number.

#### **ADD**

Adds the NVMe drive(s) for the NVMe RAID Accelerator management.

#### **REMOVE**

Removes the NVMe drive(s) from the NVMe RAID Accelerator management.

#### **Channel# ID#**

The Channel and ID of the device.

#### **ALL**

Removes all the managed NVMe drive(s) from the XLR management.

#### **LABEL**

Sets the label for the NVMe drive.

#### **LABEL#**

The label must be a maximum of 8 characters and should only contain ASCII characters.

#### **noprompt**

An optional parameter that suppresses the user confirmation prompt.

#### **nologs**

An optional parameter that suppresses log output to the log files.

#### **Examples**

```
ARCCONF NVMEDRIVECONFIG 1 ADD 0 40 LABEL Bay-1 0 41 LABEL Bay-2 nologs
ARCCONF NVMEDRIVECONFIG 1 REMOVE 0 40 nologs
ARCCONF NVMEDRIVECONFIG 1 REMOVE ALL
```

## **4.16 arccnf 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|notransfer> [length] <[> CDB> <]>
[rawhex] [noprompt] [nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#> <readwithsense|notransferwithsense> [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]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#> <read|readwithsense> [length] <[> <CDB> <]>
[rawhex|rawbinary] [noprompt] [nologs]
ARCCONF PASSTHROUGH <Controller#> <Channel# ID#> <read|readwithsense> [length] <[> <CDB> <]>
[rawhex|rawbinary] [Filename] [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.

### 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

An optional parameter that suppresses the user confirmation prompt.

**nologs**

An optional parameter that suppresses log output to the log files.

**Supported Commands**

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

**Table 4-3.** 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

**Examples**

```
ARCCONF PASSTHROUGH 1 0 1 read 512 [ 28 00 00 00 00 00 00 02 00 00 ]
ARCCONF PASSTHROUGH 1 0 1 readwithsense 512 [ 28 00 00 00 00 00 00 02 00 00 ]
ARCCONF PASSTHROUGH 1 0 1 read 512 [ 28 00 00 00 00 00 00 02 00 00 ] rawhex
ARCCONF PASSTHROUGH 1 0 1 read 512 [ 28 00 00 00 00 00 00 02 00 00 ] rawbinary
ARCCONF PASSTHROUGH 1 0 1 read 512 [ 28 00 00 00 00 00 00 02 00 00 ] rawhex C:\file.txt
ARCCONF PASSTHROUGH 1 0 1 write 512 [ 2A 00 00 00 00 00 00 02 00 00 ] C:\file.txt
ARCCONF PASSTHROUGH 1 0 1 writewithsense 512 [ 2A 00 00 00 00 00 00 02 00 00 ] C:\file.txt
ARCCONF PASSTHROUGH 1 0 1 writewithsense 512 [ 2A 00 00 00 00 00 00 02 00 00 ] /usr/file.txt
```

## 4.17 **arcconf playconfig**

### Description

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

### Notes:

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 in the following table, 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 <config xml file name> [<log file name> <FORCE [ALL | LOGICALSIZE]>]
[nologs]
```

### 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.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.18 **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

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.19 arccconf romupdate

### Description

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

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

### Syntax

```
ARCCONF ROMUPDATE <Controller#> XLR <XLRID#> <ImagePath> [noprompt] [nologs]
```

### Parameters

#### Controller#

The controller to which the update will be applied.

#### XLR

NVMe RAID Accelerator

#### XLRID#

NVMe RAID Accelerator identifier

#### ImagePath

This is the full path of the ROM image file.

#### noprompt

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

```
ARCCONF ROMUPDATE 1 XLR 0 C:\XLR.bin noprompt
```

## 4.20 **arcconf saveconfig**

### Description

**Note:** This command is supported on all OSs that support arconf/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](#) 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 in the following table, 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\Logs\saveconfig.xml.

#### LogFile

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

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.21 **arcconf savesupportarchive**

### Description

Saves configuration and status information to help diagnose a problem with your system. Saved information includes device logs, event logs, error logs, controller 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] [HostStack|GUI|Arcconf|Storlib|Hardware|Redfish] [HostStack Diagnostics] [Hardware Uartlog] [nologs]
```

### Parameters

#### Path

Path to store the log files.

#### Log type:

One of these options:

- HostStack: saves Raid stack, Driver, and Firmware logs

- Type
  - Uartlog—saves controller serial logs
  - Diagnostics—saves controller debug logs
- Arcconf: saves Arcconf logs
- Storlib: saves StorLib logs
- Hardware: saves Hardware logs
- GUI: saves GUI logs
- Redfish: saves Redfish logs

#### nologs

An optional parameter that suppresses log output to the log files.

#### Examples

```
ARCCONF SAVESUPPORTARCHIVE C:\\Adaptec\\maxView HostStack
ARCCONF SAVESUPPORTARCHIVE /var/log/maxView Storlib
ARCCONF SAVESUPPORTARCHIVE
ARCCONF SAVESUPPORTARCHIVE .
ARCCONF SAVESUPPORTARCHIVE c:\\ hardware uartlog
ARCCONF SAVESUPPORTARCHIVE hostStack diagnostics
```

## 4.22 arcconf security

### Description

Operations for SPDm certificate feature for controller. Some controllers do not support this operation.

### Syntax

```
ARCCONF SECURITY <Controller#> SPDm GETCERTIFICATE SLOT <slot# | all> [filename] [nologs]
ARCCONF SECURITY <Controller#> SPDm SETCERTIFICATE SLOT <slot# | all> <filename> [nologs]
ARCCONF SECURITY <Controller#> SPDm INVALIDATE SLOT <slot# | all> [nologs]
```

### Parameters

#### SPDM

Security Protocol and Data Model.

#### GETCERTIFICATE

Retrieves and stores certificate chains from specific slot.

#### SETCERTIFICATE

Import a certificate chain to a slot.

#### INVALIDATE

Invalidate certificate chain on the slot.

#### SLOT

SPDM certificate chain slots.

- slot#: SPDM certificate slot ID. Valid range is 0–7.
- all: Will apply the operation to all SPDM certificate chain slots.

#### filename

Absolute file path of the certificate chain to set or get on the controller.

#### nologs

An optional parameter that suppresses log output to the log files.



## Examples

```
ARCCONF SECURITY 1 SPDM GETCERTIFICATE SLOT 0 spdm_certificate_slot_0.bin
ARCCONF SECURITY 1 SPDM GETCERTIFICATE SLOT ALL
ARCCONF SECURITY 1 SPDM SETCERTIFICATE SLOT 0 spdm_certificate_slot_0.bin
ARCCONF SECURITY 1 SPDM SETCERTIFICATE SLOT ALL spdm_certificate_slot_0.bin
ARCCONF SECURITY 1 SPDM INVALIDATE SLOT 0
ARCCONF SECURITY 1 SPDM INVALIDATE SLOT ALL
```

## 4.23 arccnf setarrayparam

### Description

Changes a parameter of an array.

### Syntax

```
ARCCONF SETARRAYPARAM <Controller#> <Array#> SPARETYPE <Type> [noprompt] [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.

#### noprompt

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

```
ARCCONF SETARRAYPARAM 1 0 SPARETYPE 1
```

## 4.24 arccnf setcache

### Description

Changes a device's cache mode. Changes a controller's cache setting.

### Syntax

```
ARCCONF SETCACHE <Controller#> DRIVEWRITECACHEPOLICY <drive Type> <drive cache policy>
[noprompt] [nologs]
```

### Parameters

#### Controller#

The controller number

#### Drive Type

Configured—drive write cache policy for configured drives

#### Drive Cache Policy

Write cache policy setting; choose any of the following values

- 0—default
- 1—enable
- 2—disable
- 3—Unchanged

#### **noprompt**

An optional parameter that suppresses the user confirmation prompt.

#### **nologs**

An optional parameter that suppresses log output to the log files.

### **Examples**

```
ARCCONF SETCACHE 1 DRIVEWRITECACHEPOLICY Configured 0
```

## **4.25 arccnf setconfig**

### **Description**

Clears the controller configuration. Logical drives are deleted, hard disks are reset to the READY state.

### **Syntax**

```
ARCCONF SETCONFIG <Controller#> <DEFAULT> [noprompt] [nologs]
```

### **Parameters**

#### **Controller#**

The controller number.

#### **Default**

Clears the controller's configuration. Logical devices and arrays are deleted, and hard drives are reset to the ready state.

#### **noprompt**

An optional parameter that suppresses the user confirmation prompt.

#### **nologs**

An optional parameter that suppresses log output to the log files.

### **Examples**

```
ARCCONF SETCONFIG 1 DEFAULT
```

## **4.26 arccnf setcontrollerparam**

### **Description**

Changes a parameter of a controller.

### **Syntax**

```
ARCCONF SETCONTROLLERPARAM <Controller#> SPAREACTIVATIONMODE <Mode> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> ELEVATORSORT <Enable | Disable> [nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> OOBINTERFACE <OOBinterface>
ARCCONF SETCONTROLLERPARAM <Controller#> OOBINTERFACE PBSI I2CADDRESS <i2cAddress>
I2CCLOCKSPD <i2cClockSpeed> I2CCLOCKSTRETCH <i2cClockStretch>[nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> OOBINTERFACE MCTP I2CADDRESS <i2cAddress>
SMBUSDEVICETYPE <SMBusDeviceType> SMBUSCHANNEL <SMBusChannel> STATICEID <StaticEID>
VDMNOTIFICATION <VDMNotification>[nologs]
ARCCONF SETCONTROLLERPARAM <Controller#> UEFIHEALTHREPORTINGMODE <Enable/Disable> [nologs]
```

## Parameters

### Controller#

Controller number

### 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 elevator sort for the controller.

- Enable—Enables elevator sort
- Disable—Disables elevator sort

### OOBINTERFACE

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

- PBSI: Sets the Out-of-band interface as PBSI.
- MCTP: Sets the Out-of-band interface as MCTP.
- DISABLE: Sets the Out-of-band interface as Disable.
- RESET: Resets the Out-of-band interface.

### PBSI Parameters

PBSI Parameters	Description
I2CAddress	Sets the I <sup>2</sup> C Address of the controller. Hexadecimal input from range of 0x00–0xFF
I2CClockSpeed	Sets I <sup>2</sup> C clock speed <ul style="list-style-type: none"> <li>• 0—I<sup>2</sup>C clock speed disable (Default)</li> <li>• 1—I<sup>2</sup>C clock speed 10 kHz</li> <li>• 2—I<sup>2</sup>C clock speed 100 kHz</li> <li>• 3—I<sup>2</sup>C clock speed 400 kHz</li> <li>• 4—I<sup>2</sup>C clock speed 800 kHz</li> </ul>
I2CClockStretch	Sets the I <sup>2</sup> C clock stretch <ul style="list-style-type: none"> <li>• Enable—Enables I<sup>2</sup>C Clock Stretching</li> <li>• Disable—Disables I<sup>2</sup>C Clock Stretching</li> </ul>

### MCTP Parameters

MCTP parameters	Description
I2CAddress	Sets the I <sup>2</sup> C address of the controller. Hexadecimal input, from range of 0x00 - 0xFF. For valid address, see the Management Component Transport Protocol (MCTP) SMBus/I <sup>2</sup> C 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</li> <li>• 2—Address Resolution Protocol (ARP)</li> </ul>

MCTP parameters	Description
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) usage during initialization <ul style="list-style-type: none"> <li>• Enable—Enables Static EID</li> <li>• Disable—Disables Static EID</li> </ul>
VDMNotification	Sets the Vendor Defined Message (VDM) discovery notification <ul style="list-style-type: none"> <li>• Enable—Enables VDM discovery notification</li> <li>• Disable—Disables VDM discovery notification</li> </ul>

**UEFIHEALTHREPORTINGMODE**

UEFI Health Reporting mode setting for the controller.

Enable—Enables UEFI Health Reporting mode. All the errors related to the controller configuration is reported in UEFI.

Disable—Disables UEFI Health Reporting mode. No errors related to the controller configuration is reported in UEFI.

**noprompt**

An optional parameter that suppresses log output to the log files.

**nologs**

An optional parameter that suppresses the user confirmation prompt.

**Examples**

```
ARCCONF SETCONTROLLERPARAM 1 SPAREACTIVATIONMODE 0
ARCCONF SETCONTROLLERPARAM 1 ELEVATORSORT disable
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE Disable
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE Reset
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE PBSI I2CAddress 0xD0 I2CCLOCKSPEED 3
I2CCLOCKSTRETCH Enable
ARCCONF SETCONTROLLERPARAM 1 OOBINTERFACE MCTP I2CAddress 0xD0 SMBUSDEVICETYPE 1 SMBUSCHANNEL
Enable STATICEID Enable VDMNotification Enable
ARCCONF SETCONTROLLERPARAM 1 UEFIHEALTHREPORTINGMODE Enable
```

## 4.27 arccnf setname

### Description

Renames a logical drive.

**Note:** ARCCONF does not allow a duplicate name for logical devices in the configuration.

### Syntax

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

### Parameters

#### Controller#

Controller number

#### LogicalDrive#

Logical device ID 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.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

```
ARCCONF SETNAME 1 LOGICALDRIVE 0 LD123
```

## 4.28 arccnf setpriority

### Description

Changes a controller's rebuild or expand priority.

### Syntax

```
ARCCONF SETPRIORITY <Controller#> REBUILD <New Priority> [nologs]
ARCCONF SETPRIORITY <Controller#> 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. Rebuild priority will default to MEDIUM if the new priority specified is not supported.

#### EXPAND

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

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.29 **arccnf setstate**

### Description

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

### Syntax

```
ARCCNF SETSTATE <Controller#> DEVICE <Channel# ID#> <State> [noprompt] [nologs]
ARCCNF SETSTATE <Controller#> DEVICE <Channel# ID#> <State> [ARRAY <AR#> [AR#] ... ]
[SPARETYPE <Type>] [noprompt] [nologs]
ARCCNF SETSTATE <Controller#> LOGICALDRIVE <ld#> OPTIMAL [noprompt] [nologs]
```

### Parameters

#### Controller#

The controller number

#### Channel# ID#

The Channel and ID of the device whose state will be altered.

#### ld#

Logical drive number.

#### AR#

The array to which the device belongs to.

#### 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.

#### OPTIMAL

Force a logical device to Optimal state.

#### SPARETYPE

Sets the sparetype for the array.

- 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.

#### noprompt:

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

```
ARCCNF SETSTATE 1 DEVICE 0 0 HSP ARRAY 0
ARCCNF SETSTATE 1 DEVICE 0 0 DDD
ARCCNF SETSTATE 1 DEVICE 0 0 RDY
ARCCNF SETSTATE 1 DEVICE 0 0 HSP ARRAY 0 SPARETYPE 2
ARCCNF SETSTATE 1 LOGICALDRIVE 1 OPTIMAL
```

## 4.30 **arccnf splitmirror**

### Description

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

### Syntax

```
ARCCNF SPLITMIRROR <Controller#> ARRAY <Array#> SPLITWITHBACKUP [noprompt] [nologs]
ARCCNF SPLITMIRROR <Controller#> ARRAY <Array#> REMIRROR [noprompt] [nologs]
ARCCNF SPLITMIRROR <Controller#> ARRAY <Array#> ROLLBACK [noprompt] [nologs]
ARCCNF SPLITMIRROR <Controller#> ARRAY <Array#> ACTIVATEBACKUP [noprompt] [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 characteristic:

- If the original array contained RAID 1 or RAID 10 drives, the primary array will contain RAID 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.



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.

#### noprompt

An optional parameter that suppresses the user confirmation prompt.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 4.31 arccconf task

### Description

Performs a task on a logical drive or physical drive. Uninitializes physical drives on a controller.

### Syntax:

```
ARCCONF TASK START <Controller#> DEVICE <Channel# ID#> <Task> [PATTERN <erasePattern>]
[Unrestricted] [noprompt] [nologs]
ARCCONF TASK STOP <Controller#> DEVICE <Channel# ID#> [nologs]
```

### Parameters

#### Controller#

The controller number

#### Channel# ID#

The Channel and ID of the physical device on which task has to be performed.

### Tasks

**secureerase**—Performing a secure erase on a physical device, all data is completely and irretrievably eradicated.

### Options:

- Physical device options:
  - PATTERN <erasePattern>—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.
  - Sanitize erase patterns
    - 4—Crypto Scramble Sanitize Method - HDDs and SSDs. Causes the physical device to change encryption keys to prevent correct decryption of previously stored information, which may cause protection information, if any, to be indeterminate.
    - 5—Block Erase Sanitize Method - SSDs only. Erase voltage is applied to all NAND cells.
  - Unrestricted—Sanitize erase with this option makes the physical device available for configuration once sanitize erase fails/could not complete as well. If the user does not provide this option, default value is set to restricted. With default option, if sanitize erase fails, the only operation allowed is to start another sanitize else return the drive if it is under warranty. Applicable only to the sanitize erase methods (Crypto Scramble/Block Erase).

### noprompt

An optional parameter that suppresses the user confirmation prompt.

### nologs

An optional parameter that suppresses log output to the log files.

### Examples

```
ARCCONF TASK START 1 DEVICE 0 0 SECUREERASE PATTERN 1
ARCCONF TASK STOP 1 DEVICE 0 0
```



## 4.32 **arccconf uninit**

### Description

Firmware reserves 32 MiB at the end of each disk for storing configuration information. Storing the configuration on the drives allows for controller replacement or drive movement without configuration loss. Uninitializing the drives clears the Smart configuration (Reserve Information Sector) data and allows the drive to be exposed to the operating system.

### Syntax:

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

### Parameters

#### Controller#

Controller number.

#### Channel# ID#

The Channel and ID of the physical device which will be uninitialized.

#### ALL

Uninitializes all physical devices on the controller.

#### nologs

An optional parameter that suppresses log output to the log files.

### Examples

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

## 5. 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 in [UEFI ARCCONF Commands](#). 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:
  - a. Boot the machine to the UEFI shell prompt.
  - b. Type: `drivers`
  - c. Verify that 'MSCC UEFI Driver (version)' is listed.

### Starting UEFI ARCCONF

To start UEFI, use one of the following methods:

- Using Standalone ARCCONF in the UEFI shell
- Using UEFI ARCCONF

To start UEFI using the Standalone ARCCONF in the UEFI Shell, perform the following steps:

1. Transfer the UEFI Standalone ARCCONF (`arccconf_x64.efi`) to a portable storage device.
2. Boot the computer in UEFI shell prompt and connect the portable storage device to the test setup.
3. Identify the portable storage device and change directory to the path where `arccconf_x64.efi` is copied.
4. Get the build version by running the standalone `arccconf arccconf_x64.efi`.
5. To launch the Standalone ARCCONF, use the following syntax:

```
arccconf_x64.efi <command_name> <parameters> ...
```

To start UEFI ARCCONF, perform the following steps:

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

```
arccconf <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 following table 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 5-1.** UEFI ARCCONF Commands

Command	Description	Usage
CONSISTENCYCHECK	Modifies the background consistency check modes and its parameters of the controller.	see <a href="#">arccconf consistencycheck</a>

**Table 5-1. UEFI ARCCONF Commands (continued)**

Command	Description	Usage
CREATE	Creates a new logical drive; optionally, enables logical drive read caching, write caching.	See <a href="#">arccnf create</a>
DELETE	Deletes one or more logical devices or arrays from the controller.	See <a href="#">arccnf delete</a>
GETCONFIG	Prints the controller configuration.	See <a href="#">arccnf getconfig</a>
GETLOGS	Provides access to controller status, event logs, and usage statistics.	See <a href="#">arccnf getlogs</a>
GETSMARTSTATS	Obtains SMART statistics of all the hard drives attached to the controller.	See <a href="#">arccnf getsmartstats</a>
GETSTATUS	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.	See <a href="#">arccnf getstatus</a>
GETVERSION	Lists version information for all controllers or a specific controller's software components.	See <a href="#">arccnf getversion</a>
IDENTIFY	Identifies a physical device by blinking its LEDs.	See <a href="#">arccnf identify</a>
IMAGEUPDATE	Allows new firmware to be flashed to the hard drive.	See <a href="#">arccnf imageupdate</a>
LIST	Lists controllers in the system.	See <a href="#">arccnf list</a>
NVMEDRIVECONFIG	Manages the NVMe drives for NVMe RAID Accelerator.	See <a href="#">arccnf nvmedriveconfig</a>
PASSTHROUGH	Sends SCSI passthrough protocol (SCSI) function request to a disk drive.	See <a href="#">arccnf passthrough</a>
RESCAN	Check for removal of disk drives and for connection of new disk drives to the controller.	See <a href="#">arccnf rescan</a>
ROMUPDATE	Allows new firmware to be flashed to the controller. A reboot is required for the new firmware to take effect.	See <a href="#">arccnf romupdate</a>
SAVESUPPORTARCHIVE	Saves configuration and status information. Usage: SAVEDUPPORTARCHIVE [Path] [HostStack Hardware] [HostStack Diagnostics] [Hardware Uartlog] [nologs]	See <a href="#">arccnf savesupportarchive</a>
SECURITY	Operations for SPDm certificate feature for controller. Some controllers do not support this operation.	See <a href="#">arccnf security</a>
SETARRAYPARAM	Changes a parameter of an array.	See <a href="#">arccnf setarrayparam</a>
SETCACHE	Changes a device's cache mode. Changes a controller's cache setting.	See <a href="#">arccnf setcache</a>
SETCONFIG	Clears the controller configuration.	See <a href="#">arccnf setconfig</a>
SETCONTROLLERPARAM	Changes a parameter of a controller.	See <a href="#">arccnf setcontrollerparam</a>
SETSTATE	Changes the state of a physical device or logical device.	See <a href="#">arccnf setstate</a>
SETNAME	Renames a logical drive.	See <a href="#">arccnf setname</a>
SETPRIORITY	Changes a task's execution priority or a controller's global background task priority.	See <a href="#">arccnf setpriority</a>

**Table 5-1. UEFI ARCCONF Commands (continued)**

Command	Description	Usage
SPLITMIRROR	Splits an array consisting of one or more RAID 1 or RAID 10 logical devices into two new arrays with identical contents.	See <a href="#">arccnf splitmirror</a>
TASK	Performs a task on a logical drive, physical drive, or an array. Uninitializes physical drives on a controller.	See <a href="#">arccnf task</a>
UNINIT	Uninitializes one or more physical drives. The uninitialize command clears the meta-data from a drive; existing data on the drive is destroyed.	See <a href="#">arccnf uninit</a>

## 6. Controller Device Status

The following tables provide the status and their description of the controller, array, logical device, and physical device.

**Table 6-1.** Status Details of the Controller

Status	Description
Optimal	The controller is healthy.
Controller Has Incompatible Driver	The controller has an incompatible driver.
Failed	The controller is not in working condition.
Inaccessible	Controller communication failure error
Down/Offline	Controller offline error
Controller Lockup Error	Controller failed with a lockup error.
Missing SG Module	Controller missing module error. In order to configure this controller, you must load the SCSI Generic (SG) module.
Controller Powered OFF	Controller power off error
SmartXLR Driver Conflict	This controller has an incompatible driver.
Unknown	Controller unknown error

**Table 6-2.** Status Details of the Array

Status	Description
Ok	The array has all the logical device(s) in an optimal state.
Has Initializing Logical Device	One or more member logical device(s) has Rapid Parity Initialization (RPI) in progress.
Logical Devices Not Contiguous	The logical devices in this array are not in contiguous order. Perform consolidate space operation to consolidate all the free space to the end of the array.
Has Failed Physical Device	Device array has a bad or a missing physical device.
Has Failed Logical Device	One or more logical device(s) in the array has failed.
Failed	A physical device or logical device in the array has failed.
Has Erasing Drive	The array currently has a drive erase operation queued, running, stopped, or completed on a logical or a physical device.
Spare Drive Size Too Small	The array has a spare drive assigned which is smaller than the smallest data drive in the array.
Has Offline Logical Drive	The array has an offline or a data locked logical device(s).

**Table 6-3.** Status Details of the Logical Device

Status	Description
Optimal	The logical device is healthy and is readily accessible by the host.
Queued for Expansion	The logical device is queued for expansion.
Expanding	The logical device is undergoing online capacity expansion.
Ready for Recovery	The logical device is queued to be recovered from a failed physical device.
Recovering	The logical device is rebuilding a physical device from fault tolerant data.
Wrong Drive Replaced	A wrong physical device is replaced.
RPI In Progress	RPI is currently in progress on this logical device.
RPI Queued	RPI is currently queued on this logical device. It starts once other progress tasks are completed.
Unsupported on The Controller	Logical device is unsupported on this controller. Host access to this volume is denied. Logical device can still be deleted/reconfigured with data loss.
Unknown	The status of the logical device is unknown.
Erase In Progress	The logical device is offline and has the erase operation in progress.
Ejected	The logical device is offline from being ejected. Reinstall the removed physical devices.

**Table 6-3. Status Details of the Logical Device (continued)**

Status	Description
Not Yet Available	An expand, shrink, or move operation on the array is in progress. This logical device remains in this state until all expand, shrink, or move operations on this array are completed. All I/O requests sent to the logical drive in this state will be rejected.
Not Configured	The logical device is not yet configured.
Interim Recovery	The logical device has a bad or missing drive. Logical device is operating with reduced performance and a further physical drive failure may result in data loss depending on the fault tolerance. To correct this problem, check the data and power connections to the physical drives or replace the failed drive.
Failed	The logical device has bad or missing physical device(s).
Disabled From SCSI ID Conflict	A conflict with an existing SCSI ID exists. Check all the SCSI components to make sure they all have a unique SCSI ID.
Drive Improperly Connected	A physical device is not properly connected.
Hardware Has Overheated	A physical device temperature has crossed the threshold value.
Hardware Is Overheating	A physical device temperature is about to reach the threshold value.
Optimal (Background Parity Initialization)	Logical device is undergoing the Parity Initialization in background.
Rapid Parity Initialization	Logical device is undergoing the RPI and may not be available until it is completed.
Logical Device Reconfiguring	Logical device is reconfiguring.

**Table 6-4. Status Details of the Physical Device**

Status	Description
Ready	The physical device is readily available for RAID configuration.
Optimal	The physical device is part of an array/logical device.
Waiting For Rebuild	The physical device is waiting to be rebuilt.
Rebuilding	The data on the physical device is being rebuilt. The physical device is accessible. But the performance will be less than optimal during the rebuilding process.
Queued For Erase	The physical device is currently queued for erase and it is not available for use until the erase operation is completed.
Erase In Progress	The physical device is currently being erased and it is not available for use until the erase operation is completed.
Erase Completed	The erase process has been completed on the physical device and the physical device is offline. The physical device may now be brought online through the initialize operation.
Erase Failed	The physical device erase process is failed and the device is offline. The physical device may now be brought online through the initialize operation.
Erase Aborted	The physical device is offline due to an aborted erase process.
Predictive Failure	This physical device is predicted to fail soon. Backup all the data on the drive and replace the drive.
Transient Data Drive	The physical device is in transition from being a member of an array to being an unassigned physical device as a result of shrink array/move array operation.
Failed	The physical device is bad or missing.
Failed Due To Predictive Spare Activation	The physical device has been failed by the controller after completing a predictive spare activation.
Unsupported	The physical device is not supported by the controller.
Not Supported	The controller firmware version does not support this physical device. Replace the physical device with the one supported by the controller.
Dedicated Hot Spare	A dedicated hot spare is assigned to one or more arrays.
Auto Replace Hot Spare	An auto-replace hot spare is assigned to a specific array. After using an auto-replace spare to rebuild a failed logical drive, it becomes a permanent part of the array.
Raw	A physical device is in RAW state which has no or unknown file system.
Size Not Valid	The physical device size is not valid.

## 7. Controller, Array, and Physical Device

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

**Table 7-1.** 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.
Vendor ID	Vendor ID of the controller
Accelerator ID	Accelerator ID of the controller
Device ID	Device ID of the controller
Subsystem Vendor ID	Subsystem Vendor ID of the controller
Subsystem Device ID	Subsystem Device ID of the controller
Accelerator Serial Number	A unique number assigned to the accelerator, used for identification and inventory purposes.
Accelerator World Wide Name	A World Wide Name (WWN) is an unique identifier of the accelerator.
Physical Slot	PCI slot number to which the controller is connected.
Negotiated PCIe Data Rate	Negotiated PCIe data-rate of the controller.
PCI Address (Domain:Bus:Device:Function)	PCI address describes the PCI address for the controller.
Defunct disk drive count	Number of failed drive.
NCQ status	Native Command Queuing, or NCQ, lets NVMe drives arrange commands into the most efficient order for optimum performance.
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 minutes. This option is primarily used to tune the controller's performance for video applications.
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.
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.
NVMe <sup>®</sup> Supported	Describes whether the controller supports NVMe devices in the configuration.
NVMe Configuration Supported	Describes whether NVMe devices can be used for configuring logical devices.
NVRAM Checksum Status	Describes the NVRAM Checksum status.
Reboot Required Reasons	Describes reasons why the controller needs a reboot.

**Table 7-1. Controller (continued)**

Properties	Description
Configured Drives	<p>Sets the write cache policy for the configured physical devices on the controller.</p> <ul style="list-style-type: none"> <li>• Default: Allows the controller to control the drive write cache policy of all configured physical devices.</li> <li>• Enabled: The drive write cache for the physical device will be enabled by the controller. Setting to enabled can increase write performance but risks losing the data in the cache on sudden power loss to all configured physical devices.</li> <li>• Disabled: The drive write cache for the physical devices will be disabled by the controller.</li> <li>• Unchanged: Sets the physical devices factory default policy for all configured drives.</li> </ul>
Logical devices/Failed/Degraded	Total number of logical devices/number of logical devices failed/number of logical devices degraded.
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.
UEFI Health Reporting Mode	<p>Indicates the UEFI Health Reporting mode of the controller. The available options are Enable/Disable:</p> <ul style="list-style-type: none"> <li>• Enabling the mode allows reporting of all the errors related to the controller configuration in UEFI.</li> <li>• Disabling the mode stops reporting of errors related to the controller configuration in UEFI.</li> </ul>
Background consistency check	<p>If your controller supports background consistency check, controller firmware continually and automatically checks your logical drives for bad or inconsistent data when they are in use.</p> <p>When supported, the controller enables this check by default and runs during idle periods.</p> <p>When enabled, Consistency Check performs a background check on logical drives every 14 days from the time the last check was completed. Factors that may extend this time include the Priority mode, parallel count, number of logical devices, and host I/O activity.</p>
Consistency Check Delay	<p>Determines the time interval for which a controller must be inactive before a consistency check is initiated on the logical drives. The value can be between 0 and 30 to specify the duration of the delay in seconds. A value of 0 disables the scan. Controller default value is 3 seconds.</p>
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.
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.



**Table 7-1. Controller (continued)**

Properties	Description
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.
CPLD Revision	Describes the controller CPLD revision.
Hardware Revision	Describes the hardware revision information about the controller.
Hardware Minor Revision	Describes the hardware minor revision of the controller.
OOB Interface	Out of Band Interface setting.
Pending OOB Interface	Out of Band Interface is in pending state, reboot the system for the new setting to be applied.
I <sup>2</sup> C Address	The Inter-Integrated Circuit (I <sup>2</sup> C) slave address.
Pending I <sup>2</sup> C Address	I <sup>2</sup> C address is in pending state, reboot the system for the new setting to be applied.
<b>PDM Related Properties</b>	
Version	SPDM version
Endpoint ID	Endpoint identifier
Crypto Timeout Exponent	Time requires for cryptographic processing
Authority Key ID	Authority key identifier
Slot [0...7]	Slot identifier
<b>PBSI Related Properties</b>	
I <sup>2</sup> C Clock Speed	I <sup>2</sup> C clock speed. Value in KHz.
Pending I <sup>2</sup> C Clock Speed	I <sup>2</sup> C clock speed is in pending state, reboot the system for the new setting to be applied.
I <sup>2</sup> C Clock Stretching	I <sup>2</sup> C clock stretching is enabled or disabled.
Pending I <sup>2</sup> C Clock Stretching	I <sup>2</sup> C clock stretching is in pending state, reboot the system for the new setting to be applied.
<b>Capabilities Related Properties</b>	
Cache Negotiated State	Responder supports the ability to cache the Negotiated state across a Reset.
Digests and Certificate	Responder supports digests and certificate messages.
Challenge	Responder supports challenge request message.
Measurements Without Signature	Responder supports measurements but cannot perform signature generation.
Measurements With Signature	Responder supports measurements and can perform signature generation.
Measurements Fresh	Responder can recompute all measurements in a manner that is transparent to the rest of the system.
Single Pre-Shared Key	The Responder supports one pre shared key.
Derived Pre-Shared Key	The Responder supports derived pres shared key.
<b>MCTP Related Properties</b>	
SMBus Device Type	System Management (SM) Bus Device Type
SMBus Channel	System Management (SM) Bus Channel setting
Static EIDs Use On Initialization	Static End Point Identifiers (EID) usage during initialization
VDM Notification	Vendor Defined Message (VDM) discovery notification
Pending SMBus Device Type	SMBus Device Type is in pending state, reboot the system for the new setting to be applied.

**Table 7-1. Controller (continued)**

Properties	Description
Pending SMBus Channel:	SMBus Channel is in pending state, reboot the system for the new setting to be applied.
Pending Static EIDs Use On Initialization	Static EIDs Use On Initialization is in pending state, reboot the system for the new setting to be applied.
Pending VDM Notification	VDM Notification is in pending state, reboot the system for the new setting to be applied.

**Table 7-2. 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.
Interface	Disk drives that are the member of array have interface type NVMe <sup>®</sup> . 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.
Associated Split Mirror Array	Name of its associated array, if array is of type split mirror.
Transformation Status	Indicates if any logical drive of the array is undergoing transformation.
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.

**Table 7-3. 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 in Operation System.
Disk Name	Name of the logical disk drive.
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.
Status of Logical Device	Status of logical device based on health of RAID members of logical device.
Parity Initialization Status	RPI status of this logical device.
Size	Logical device size.
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 devices, excluding parity-only devices.
Interface Type	Disk drives which are the member of logical have interface type NVMe <sup>®</sup> .
Device Type	Indicates logical drive is of type data or cache.

**Table 7-3. Logical Device (continued)**

Properties	Description
Heads	Indicates the predefined 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.
Mount Points	Describes the operating system device names of the logical device and vice versa.
Volume Unique Identifier	Logical device unique string identifier.
<b>Consistency Check Information</b>	
Consistency Check Status	Indicates if a consistency check is currently in progress or not running. It is not applicable for non-redundant logical device.
Last Consistency Check Completion Time	Indicates the date and time of the last consistency check that occurred in the logical drive.
Last Consistency Check Duration	Indicates how long the last consistency check took for that logical device (seconds).

**Table 7-4. 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".
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 in Operating System.
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).
Physical Block Size	It indicates the physical block size in bytes. Valid only for physical drives (NVMe®).
Dedicated Spare for drive	If drive is acting as spare.
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.
PCI Address (Domain:Bus:Device.Function)	Physical location information of the drive that contains domain, bus, device, and function.
Parent PCI Address (Domain:Bus:Device.Function)	Parent physical location information of the drive that contains domain, bus, device, and function.
XLR Configuration State	Current XLR configuration state of the physical device based on the operations done on it.
Label	The label for the NVMe drive.
Array	Specifies the array ID, where the drive is constituent as a data or spare drive.
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.
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.
Write Cache	Indicates the write cache policy on the device.
S.M.A.R.T.	SMART supported.

**Table 7-4. Physical Device (continued)**

Properties	Description
S.M.A.R.T. warnings	Any SMART error reported in the physical device.
Unsupported Reason(s)	Describes reasons why the physical drive is unsupported.
SSD	If drive type is SSD.
Current Temperature	Current temperature of the physical device.
Maximum Temperature	The maximum temperature reported by the physical device.
Threshold Temperature	The threshold 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.
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.
Power On Hours	The number of operational power-on hours of SSD.
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.
Interface Type	Interface type supported by the physical device.
<b>Drive PHY information</b>	
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.
<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
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

**Table 7-4.** Physical Device (continued)

Properties	Description
Sectors Reads	Sectors Reads
Sectors Written	Sectors Written
Service Hours	Service Hours
Marked Bad Blocks	Marked Bad Blocks

## 8. Revision History

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

Revision	Date	Description
B	03/2025	<p>The following is the summary of changes in revision B of this document:</p> <ul style="list-style-type: none"> <li>• Added Windows® entries for ARCCONF execution and batch environment</li> <li>• Added new section—arccconf modify, arccconf security</li> <li>• Updated the syntax/examples of—arccconf getstatus, arccconf nvmedriveconfig, arccconf savesupportarchive, arccconf setcontrollerparam, and arccconf task</li> <li>• Updated description of—arccconf getconfig</li> <li>• Updated parameters in—arccconf savesupportarchive (Log type) and arccconf task (Options)</li> <li>• Added parameters in—arccconf getconfig (OOB), arccconf getstatus (Channel# ID#, LogicalDrive#, ALL), arccconf nvmedriveconfig (ALL, Channel# ID#), and arccconf setcontrollerparam (OOBINTERFACE)</li> <li>• Removed parameters in—arccconf nvmedriveconfig (#PCIADDRESS) and arccconf setcontrollerparam (SANITIZELOCK)</li> <li>• Updated the following tables—UEFI ARCCONF Commands table, Controller table, and Physical Devices table</li> </ul>
A	12/2024	Initial revision

# Microchip Information

## Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>.

ISBN: 979-8-3371-0929-9

## Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at [www.microchip.com/en-us/support/design-help/client-support-services](http://www.microchip.com/en-us/support/design-help/client-support-services).

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

## Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.