



3ware®

Serial ATA RAID Controller

Command Line Interface

Supports the 9590SE-4ME for the Power Mac® G5

PN: 720-0140-00
September 2006

CLI Guide for the Power Mac G5

Copyright

©2003-2006 Applied Micro Circuits Corporation (AMCC). All rights reserved. This publication may be copied or reproduced for reference purposes only. All other purposes require the express written consent of AMCC, 215 Moffett Park Drive, Sunnyvale, CA 94089. AMCC shall not be responsible or liable for, and shall be held harmless against, any and all damages, claims, and/or disputes that arise from the copying or reproduction of this publication.

Trademarks

3ware®, Escalade®, 3DM®, and TwinStor® are all registered trademarks of AMCC. The 3ware logo, 3BM , StorSwitch , and R5 Fusion are all trademarks of AMCC. Apple®, the Apple logo, and PowerMac® are trademarks of Apple Computer Inc., registered in the United States and/or other countries. Safari is a trademark of Apple Computer, Inc. PowerPC and the PowerPC logo are trademarks of International Business Machines Corporation. Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both. Windows® is a registered trademark of Microsoft Corporation in the United States and other countries. Firefox® is a registered trademark of the Mozilla Foundation. PCI Express® is a registered trademark of PCI-SIG®. All other trademarks herein are property of their respective owners.

Disclaimer

While every attempt is made to make this document as accurate as possible, AMCC assumes no responsibility for errors or omissions in this document, nor does AMCC make any commitment to update the information contained herein.

Table of Contents

Chapter 1. Introduction to the 3ware Command Line Interface	3
Features of the CLI	3
Supported Operating Systems	4
Terminology	4
Installing the 3ware CLI on Mac OS X	5
Working with 3ware CLI	5
Using the command interface interactively	6
Using a single command with output	6
Using an input file to execute a script	7
Outputting the CLI to a Text File	7
Conventions	8
Understanding RAID Levels and Concepts	8
RAID Concepts	9
Available RAID Configurations	10
Determining What RAID Level to Use	13
Chapter 2. CLI Syntax Reference	15
Common Tasks Mapped to CLI Commands	15
Syntax Overview	18
Shell Object Commands	19
focus <i>Object</i>	19
show	20
show ver	21
show alarms [reverse]	22
show diag	22
show rebuild	22
show verify	23
show selftest	23
flush	24
rescan	24
commit	24
update fw= <i>filename_with_path</i> [force]	24
Controller Object Commands	25
/cx show	26
/cx show <i>attribute</i> [<i>attribute ...</i>]	27
/cx show driver	28
/cx show model	28
/cx show firmware	28
/cx show bios	28
/cx show monitor	28
/cx show serial	29
/cx show pcb	29
/cx show pchip	29
/cx show achip	29

/cx show numdrives	.29
/cx show numports	.30
/cx show numunits	.30
/cx show ctlbus	.30
/cx show exportjbod	.30
/cx show spinup	.31
/cx show stagger	.31
/cx show ondegrade	.31
/cx show autocarve	.31
/cx show carvesize	.32
/cx show memory	.32
/cx show autorebuild	.33
/cx show unitstatus	.33
/cx show allunitstatus	.34
/cx show drivestatus	.34
/cx show all	.35
/cx add type=<RaidType> disk=<p:-p> [stripe=Stripe] [noscan] [group=<3/4/5/6>] [nocache] [autoverify] [ignoreECC] [name=string] [storsave=<protect balance perform>]	.36
/cx rescan [noscan]	.38
/cx commit	.39
/cx flush	.39
/cx update fw=filename_with_path [force]	.39
/cx show alarms [reverse]	.40
/cx show diag	.40
/cx show rebuild	.41
/cx show verify	.43
/cx show selftest	.44
/cx add rebuild=ddd:hh:duration	.45
/cx add verify=ddd:hh:duration	.46
/cx add selftest=ddd:hh	.47
/cx del rebuild=slot_id	.47
/cx del verify=slot_id	.48
/cx del selftest=slot_id	.48
/cx set rebuild=enable disable 1..5	.48
/cx set verify=enable disable 1..5	.49
/cx set selftest=enable disable [task=UDMA SMART]	.49
/cx set exportjbod=on off	.49
/cx set ondegrade=cacheoff follow	.50
/cx set spinup=nn	.50
/cx set stagger=nn	.50
/cx set autocarve=on off	.50
/cx set carvesize=[1024..2048]	.51
/cx set autorebuild=on off	.51
/cx start mediascan	.52
/cx stop mediascan	.52
Unit Object Commands	.53
/cx/ux show	.53
/cx/ux show attribute [attribute ...]	.54
/cx/ux show status	.54
/cx/ux show rebuildstatus	.55
/cx/ux show verifystatus	.55
/cx/ux show initializestatus	.55

/cx/ux show name	.55
/cx/ux show serial	.55
/cx/ux show qpolicy	.56
/cx/ux show storsave	.56
/cx/ux show identify	.56
/cx/ux show autoverify	.56
/cx/ux show cache	.57
/cx/ux show ignoreECC	.57
/cx/ux show volumes	.57
/cx/ux show all	.57
/cx/ux remove [noscan] [quiet]	.58
/cx/ux del [noscan] [quiet]	.58
/cx/ux start rebuild disk=p<p:-p...> [ignoreECC]	.59
/cx/ux start verify	.59
/cx/ux pause rebuild	.60
/cx/ux resume rebuild	.60
/cx/ux stop verify	.60
/cx/ux flush	.61
/cx/ux set autoverify=on/off	.61
/cx/ux set cache=on/off [quiet]	.61
/cx/ux set identify=on/off	.62
/cx/ux set ignoreECC=on/off	.62
/cx/ux set name=string	.62
/cx/ux set qpolicy=on/off	.62
/cx/ux set storsave=protect balance perform [quiet]	.63
/cx/ux migrate type=RaidType [disk=p:-p] [group=3 4 5 6] [stripe=Stripe] [noscan] [nocache] [autoverify]	.64
Port Object Commands	.69
/cx/px show	.69
/cx/px show <i>attribute</i> [<i>attribute ...</i>]	.69
/cx/px show status	.70
/cx/px show model	.70
/cx/px show serial	.70
/cx/px show firmware	.70
/cx/px show identify	.70
/cx/px show ncq	.71
/cx/px show lspeed	.71
/cx/px show capacity	.71
/cx/px show smart	.71
/cx/px show all	.72
/cx/px remove [noscan] [quiet]	.73
/cx/px set identify=on/off	.73
BBU Object Commands	.74
/cx/bbu show	.74
/cx/bbu show <i>attribute</i> [<i>attribute ...</i>]	.75
/cx/bbu show <i>status</i>	.75
/cx/bbu show <i>batinst</i>	.76
/cx/bbu show <i>lasttest</i>	.76
/cx/bbu show <i>volt</i>	.76
/cx/bbu show <i>temp</i>	.76
/cx/bbu show <i>cap</i>	.76
/cx/bbu show <i>serial</i>	.77
/cx/bbu show <i>fw</i>	.77

/cx/bbu show pcb	.77
/cx/bbu show bootloader	.77
/cx/bbu show all	.77
/cx/bbu test [quiet]	.77
/cx/bbu enable	.78
/cx/bbu disable [quiet]	.78
Enclosure Object Commands	.79
/ex show	.79
/ex show <i>attribute</i> [<i>attribute</i> ...]	.80
/ex show controllers	.80
/ex show slots	.80
/ex show fans	.81
/ex show temp	.81
/ex show all	.81
/ex/slot <i>x</i> show	.81
/ex/slot <i>x</i> show identify	.82
/ex/slot <i>x</i> set identify=on off	.82
/ex/fan <i>x</i> show	.82
/ex/temp <i>x</i> show	.82
Help Commands	.83
Help with specific commands	.83
Help with attributes	.84
help	.85
help <i>show</i>	.85
help <i>flush</i>	.85
help <i>rescan</i>	.86
help update	.86
help <i>commit</i>	.86
help <i>focus</i>	.86
help /cx	.86
help /cx/ux	.86
help /cx/px	.86
help /cx/bbu	.87
help /ex	.87
help /ex/slot <i>x</i>	.87
help /ex/fan <i>x</i>	.87
help /ex/temp <i>x</i>	.87
Return Code	.87

About this CLI Guide

3ware Serial ATA RAID Controller CLI Guide for the Power Mac G5 provides instructions for configuring and maintaining your 3ware controller using 3ware's command line interface (CLI).



Note: This guide assumes that you have already installed your 3ware RAID controller in your system and set up your 3ware Sidecar. If you have not yet done so, see the installation guide that came with your 3ware RAID controller for instructions.



Note: The 3ware controller used for the Apple® Power Mac® G5 is the 9590SE-4ME. AMCC makes a number of other 3ware controller models, including 9000-series (9500S, 9550SX, and 9590SE), and earlier 7000/8000 series boards, that work on other platforms.

The Command Syntax described in this document supports all of those boards, and some commands indicate the particular boards that support the command.

For your Power Mac G5, the CLI commands that apply to the 9590SE-4ME are those identified as 9000 series and those that are identified as 9590SE.



Note: You may notice mention of RAID 50 in the Command Syntax sections. RAID 50 requires six or more drives, and so is not available using the 9590SE-4ME.

Table 1: Sections in this CLI Guide

Chapter	Description
1. Introduction to 3ware Command Line Interface	Installation, features, concepts
2. CLI Syntax Reference	Describes individual commands using the primary syntax

There are often multiple ways to accomplish the same configuration and maintenance tasks for your 3ware controller. While this manual includes instructions for performing tasks using the command line interface, you can also use 3DM^{®2} (3ware Disk Manager) to accomplish these tasks.

For details, see the user guide or the 3ware HTML Bookshelf.

Introduction to the 3ware Command Line Interface

3ware RAID controllers can be managed using the CLI via a command line or script.

**Important!**

For all of the functions of the 3ware CLI to work properly, you must have the proper CLI, firmware, and driver versions installed. Check <http://www.3ware.com> for the latest versions and upgrade instructions.

This chapter includes the following sections:

- “Features of the CLI” on page 3
- “Installing the 3ware CLI on Mac OS X” on page 5
- “Working with 3ware CLI” on page 5
- “Understanding RAID Levels and Concepts” on page 8

Features of the CLI

3ware CLI is a command line interface for managing 3ware RAID Controllers. It provides controller, logical unit, drive, and enclosure management. It can be used in both interactive and batch mode, providing higher level API (application programming interface) functionalities.

You can use the CLI to view unit status and version information and perform maintenance functions such as adding or removing drives. 3ware CLI also includes advanced features for creating and deleting RAID units online.

For a summary of what you can do using the CLI, see “Common Tasks Mapped to CLI Commands” on page 15.

Supported Operating Systems

The 3ware CLI is supported under the following operating systems:

- **Mac OS X**, running in a Power Mac® G5 (PowerPC™-based) with PCI Express®.
- **Windows®**. Windows 2000, Windows XP, and Windows Server 2003, both 32-bit and 64-bit.
- **Linux®**. Redhat, SuSE, both 32-bit and 64-bit.
- **FreeBSD®**, both 32-bit and 64-bit.

Terminology

This document (*3ware Serial ATA RAID Controller CLI Guide for the Power Mac G5*) uses the following terminology:

Logical Units. Usually shortened to “units.” These are block devices presented to the operating system. A logical unit can be a one-tier, two-tier, or three-tier arrangement. Spare and Single logical units are examples of one-tier units. RAID 1 and RAID 5 are examples of two-tier units and as such will have sub-units. RAID 10 is an example of a three-tier unit and as such will have sub-sub-units.

Port. A controller has one or many ports. Each port can be attached to a single disk drive. On a controller such as the 9590SE-4ME, with a multilane serial port connector, one connector supports four ports.

For additional information about 3ware controller concepts and terminology, see the user guide that came with your 3ware RAID controller or the user guide portions of the 3ware HTML Bookshelf.

Installing the 3ware CLI on Mac OS X

3ware CLI can be installed from the 3ware software CD, or the latest version can be downloaded from the 3ware web site, <http://www.3ware.com/support/download.asp>.

To install 3ware CLI on Mac OS X

- Run the installer and select CLI as the software to be installed.

Permissions Required to Run CLI

To run CLI, you can be logged onto the Power Mac G5 with one of the following sets of permissions:

- Administrator
- Root

You can also use SUDO to run CLI.

To start CLI

- In a Terminal window, type

```
sudo ./tw_cli
```

If prompted, enter your password.

Working with 3ware CLI

You can work with the 3ware CLI in different ways:

- Interactively, entering commands at the main prompt
- As a series of single commands
- By creating a script—an input file with multiple commands

The next few topics shows examples of these different methods.

- “Using the command interface interactively” on page 6
- “Using a single command with output” on page 6
- “Using an input file to execute a script” on page 7
- “Outputting the CLI to a Text File” on page 7

Examples shown in the CLI Syntax Reference chapter reflects the interactive method.

Using the command interface interactively

You can use 3ware CLI interactively, entering commands at the main prompt and observing the results on the screen.

To use the CLI interactively

- 1 Enter the following command:

```
./tw_cli
```

The main prompt is displayed, indicating that the program is awaiting a command.

```
//localhost>
```

- 2 At the CLI prompt, you can enter commands to show or act on 3ware controllers, units, and drives.

For example,

```
//localhost> show
```

displays all controllers in the system and shows details about them, like this:

Ctl	Model	Ports	Drives	Units	NotOpt	RRate	VRate	BBU
c0	9590SE-4	4	4	1	0	3	5	TESTING

Using a single command with output

You can use 3ware CLI with line arguments, processing a single command at a time. To do so, simply enter the command and the arguments.

Single commands can be useful when you want to perform a task such as redirecting the output of the command to a file. It also allows you to use the command line history to eliminate some typing.

Syntax

```
tw_cli <command line arguments>
```

Example

```
tw_cli /c0 show diag > /tmp/3w_diag.out
```

Using an input file to execute a script

You can operate 3ware CLI scripts by executing a file. The file is a text file containing a list of CLI commands which you have entered in advance. Each command must be on a separate line.

Syntax

```
tw_cli -f <filename>
```

Where <filename> is the name of the text file you want to execute.

Example

```
tw_cli -f clicommand.txt
```

This example executes the file `clicommand.txt`, and runs the CLI commands included in that file.

Scripting example

Following is a scripting example using a text file called `config_unit.txt`, containing three commands. This example sets up a 4-port controller with two units, each with 2 drives mirrored. It then prints the configurations for verification. The commands included in the script file are:

```
/c0 add type=raid1 disk=0-1  
/c0 add type=raid1 disk=0-1  
/c0 show
```

Outputting the CLI to a Text File

You can have the output of the 3ware CLI, including errors, sent to a text file by adding `2>&1` to the end of the line. This could be useful, for example, if you want to email the output to AMCC Technical Support.

Examples

```
tw_cli /c2/p0 show >> controller2port0info.txt 2>&1
```

or

```
tw_cli /c0 show diag >> Logfile.txt 2>&1
```

Conventions

The following conventions are used through this guide:

- In text, `monospace font` is used for code and for things you type.
- In descriptions and explanations of commands, a bold font indicates the name of commands and parameters, for example, **/c0/p0 show all**.
- In commands, an italic font indicates items that are variable, but that you must specify, such as a controller ID, or a unit ID, for example, **/c0/p0 show *attribute***, and **/cx/px show all**
- In commands, brackets around an item indicates that it is optional.
- In commands, ellipses (...) indicate that more than one parameter at a time can be included, for example, **/c0/p0 show *attribute* [*attribute ...*]**, or that there is a range between two values from which you can pick a value, for example, **/cx set carvesize=[1024...2048]**.
- In commands, a vertical bar (|) indicates an 'or' situation where the user has a choice between more than one attribute, but only one can be specified.

Example: In the command to rescan all ports and reconstitute all units, the syntax appears as **rescan [*cid...*] [**noscan**]**. The italic *cid* indicates that you need to supply a controller ID. The ellipses (...) indicate that you can specify more than one controller ID, separated by spaces. The brackets [] indicate that you may omit the controller ID, to rescan all controllers, and the noscan parameter, so that the operation will be reported to the operating system.

Understanding RAID Levels and Concepts

3ware RAID controllers use a Redundant Array of Inexpensive Disks (RAID) to increase your storage system's performance and provide fault tolerance (protection against data loss).

This section organizes information about RAID concepts and configuration levels into the following topics:

- “RAID Concepts” on page 9
- “Available RAID Configurations” on page 10
- “Determining What RAID Level to Use” on page 13

RAID Concepts

The following concepts are important to understand when working with a RAID controller:

- **Arrays and Units.** In the storage industry, the term “array” is used to describe two or more disk drives that appear to the operating system as a single unit. When you work with 3ware software, “unit” is the term used to refer to an array of disks that is configured and managed through the 3ware software. Single-disk units can also be configured in the 3ware software.
- **Mirroring.** Mirrored arrays write data to paired drives simultaneously. If one drive fails, the data is preserved on the paired drive. Mirroring provides data protection through redundancy. In addition, mirroring using a 3ware RAID controller provides improved performance because 3ware’s TwinStor technology reads from both drives simultaneously.
- **Striping.** Striping across disks allows data to be written and accessed on more than one drive, at the same time. Striping combines each drive’s capacity into one large volume. Striped disk arrays achieve highest transfer rates and performance at the expense of fault tolerance.
- **Distributed Parity.** Parity works in combination with striping on RAID 5. Parity information is written to each of the striped drives, in rotation. Should a failure occur, the data on the failed drive can be reconstructed from the data on the other drives.
- **Hot Swap.** The process of exchanging a drive without having to shut down the system. This is useful when you need to exchange a defective drive in a redundant array.
- **Array Roaming.** The process of removing a unit from a controller and putting it back later, either on the same controller, or a different one, and having it recognized as a unit. The disks may be attached to different ports than they were originally attached to, without harm to the data.

For definitions of other terms used throughout the documentation, see the “Glossary”.

Available RAID Configurations

The following RAID levels and configurations are available for drives attached to a 9590SE-4ME 3ware RAID controller:

- RAID 0
- RAID 1
- RAID 5
- RAID 10
- Single Disk
- Hot Spare



Note: You may notice mention of RAID 50 in the Command Syntax sections of the CLI Guide. RAID 50 requires six or more drives, and so is not available using the 9590SE-4ME.

For how to configure RAID units, see “Configuring a New Unit” on page 34.

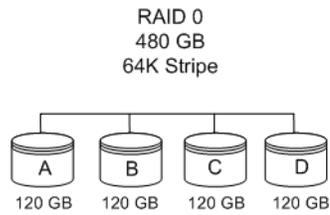
RAID 0

RAID 0 provides improved performance, but no fault tolerance. Since the data is striped across more than one disk, RAID 0 disk arrays achieve high transfer rates because they can read and write data on more than one drive simultaneously. The stripe size is configurable during unit creation. Requires a minimum of two drives.

When drives are configured in a striped disk array (see Figure 1), large files are distributed across the multiple disks using RAID 0 techniques.

Striped disk arrays give exceptional performance, particularly for data intensive applications such as video editing, computer aided design and geographical information systems.

RAID 0 arrays are not fault tolerant. The loss of any drive results in the loss of all the data in that array, and can even cause a system hang, depending on your operating system. RAID 0 arrays are not recommended for high availability systems unless additional precautions are taken to prevent system hangs and data loss.

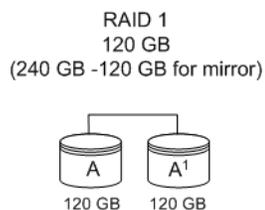
Figure 1. RAID 0 Configuration Example

RAID 1

RAID 1 provides fault tolerance and a speed advantage over non-RAID disks. Also known as a mirrored array. Mirroring is done on pairs of drives. Mirrored disk arrays write the same data to two different drives using RAID 1 algorithms (see Figure 2). This gives your system fault tolerance by preserving the data on one drive if the other drive fails. Fault tolerance is a basic requirement for critical systems like web and database servers.

3ware uses a patented technology, TwinStor®, on RAID 1 arrays for improved performance during sequential read operations. With TwinStor technology, read performance is twice the speed of a single drive during sequential read operation.

The adaptive algorithms in TwinStor technology boost performance by distinguishing between random and sequential read requests. For the sequential requests generated when accessing large files, both drives are used, with the heads simultaneously reading alternating sections of the file. For the smaller random transactions, the data is read from a single optimal drive head.

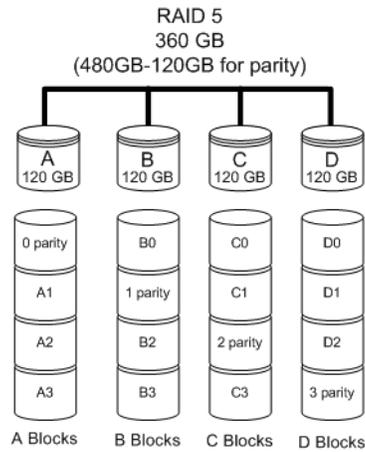
Figure 2. RAID 1 Configuration Example

RAID 5

RAID 5 provides performance, fault tolerance, high capacity, and storage efficiency. Requires a minimum of three drives. Combines striping data with parity (exclusive OR) to restore data in case of a drive failure.

Parity information is distributed across all drives rather than being concentrated on a single disk (see Figure 3). This avoids throughput loss due to contention for the parity drive.

Figure 3. RAID 5 Configuration Example



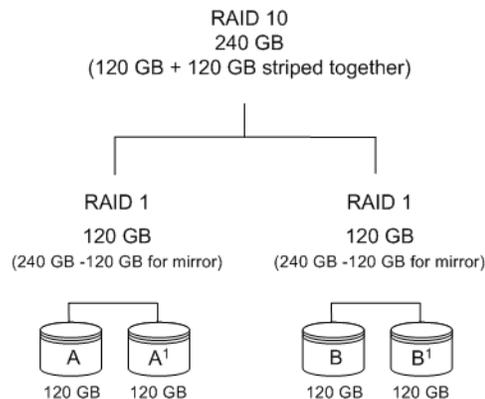
RAID 10

RAID 10 is a combination of striped and mirrored arrays for fault tolerance and high performance.

When drives are configured as a striped mirrored array, the disks are configured using both RAID 0 and RAID 1 techniques, thus the name RAID 10 (see Figure 4). A minimum of four drives are required to use this technique. The first two drives are mirrored as a fault tolerant array using RAID 1. The third and fourth drives are mirrored as a second fault tolerant array using RAID 1. The two mirrored arrays are then grouped as a striped RAID 0 array using a two tier structure. Higher data transfer rates are achieved by leveraging TwinStor and striping the arrays.

In addition, RAID 10 arrays offer a higher degree of fault tolerance than other types of RAID levels (such as RAID 1 and RAID 5), since the array can sustain multiple drive failures without data loss. Please note that if both halves of a mirrored pair in the RAID 10 array fail, then all of the data will be lost.

Figure 4. RAID 10 Configuration Example



Single Disk

A single drive can be configured as a unit through 3ware software. (3DM 2 or CLI). Like disks in other RAID configurations, single disks contain 3ware Disk Control Block (DCB) information and are seen by the OS as available units.

Single drives are not fault tolerant and therefore not recommended for high availability systems unless additional precautions are taken to prevent system hangs and data loss.

Hot Spare

A hot spare is a single drive, available online, so that a redundant unit can be automatically rebuilt in case of drive failure.

Determining What RAID Level to Use

Your choice of which type of RAID unit (array) to create will depend on your needs. You may wish to maximize speed of access, total amount of storage, or redundant protection of data. Each type of RAID unit offers a different blend of these characteristics.

The following table provides a brief summary of RAID type characteristics.

Table 2: RAID Configuration Types

RAID Type	Description
RAID 0	Provides performance, but no fault tolerance.
RAID 1	Provides fault tolerance and a read speed advantage over non-RAID disks.
RAID 5	This type of unit provides performance, fault tolerance, and high storage efficiency.
RAID 10	A combination of striped and mirrored units for fault tolerance and high performance.
Single Disk	Not a RAID type, but supported as a configuration. Provides for maximum disk capacity with no redundancy.

You can create one or more units, depending on the number of drives you have installed.

Using Drive Capacity Efficiently

To make the most efficient use of drive capacity, it is advisable to use drives of the same capacity. This is because the capacity of each drive is limited to the capacity of the smallest drive in the unit.

The total array capacity is defined as follows:

Table 3: Drive Capacity

RAID Level	Capacity
Single Disk	capacity of the drive
RAID 0	(number of drives) X (capacity of the smallest drive)
RAID 1	capacity of the smallest drive
RAID 5	(number of drives - 1) X (capacity of the smallest drive) Storage efficiency increases with the number of disks: storage efficiency = (number of drives -1)/(number of drives)
RAID 10	(number of drives / 2) X (capacity of smallest drive)

Through drive coercion, the capacity used for each drive is rounded down so that drives from differing manufacturers are more likely to be able to be used as spares for each other. The capacity used for each drive is rounded down to the nearest GB for drives under 45 GB (45,000,000,000 bytes), and rounded down to the nearest 5 GB for drives over 45 GB. For example, a 44.3 GB drive will be rounded down to 44 GB, and a 123 GB drive will be rounded down to 120 GB. For more information, see the discussion of drive coercion under “Creating a Hot Spare” on page 42.

CLI Syntax Reference



Note: The 3ware controller used for the Power Mac G5 is the 9590SE-4ME. AMCC makes a number of other 3ware controller models, including 9000-series (9500S, 9550SX, and 9590SE), and earlier 7000/8000 series boards, that work on other platforms.

The command syntax described in this document supports all of those boards, however some features are only supported on specific 3ware controller models. We have tried to indicate the controller model in the Syntax description. For example, a command that is only available for all 9000-series models will specify “9000 series” or “9000 only,” while a command that is available for only 9550SX and 9590SE models will specify “9550SX and 9590SE”, and a command that is available only on 7000/8000-series controllers will specify “7000/8000 only.”

For your Power Mac G5, the CLI commands that apply to the 9590SE-4ME are those identified as 9000 series and those that are identified as 9590SE.



Note: You may notice mention of RAID 50 in the Command Syntax sections. RAID 50 requires six or more drives, and so is not available using the 9590SE-4ME.

This chapter provides detailed information about using the command syntax for the 3ware CLI.

Throughout this chapter the examples reflect the interactive method of executing 3ware CLI.

Common Tasks Mapped to CLI Commands

The table below lists many of the tasks people use to manage their RAID controllers and units, and lists the primary CLI command associated with those tasks.

Table 4: Common Tasks Mapped to CLI Commands

Task	CLI Command	Page
Controller Configuration Tasks		
View information about a controller	/cx show	26
View controller policies	/cx show [attribute] [attribute]	27
Set policies for a controller		
• Export JBODs	/cx set exportjbod	49
• Modify staggered spinup	/cx set stagger and /cx set spinup	50
• Disable write cache on unit degrade	/cx set ondegrade	50
• Enable/disable autocarving	/cx set autocarve	50
• Enable/disable autorebuild	/cx set autorebuild	50
• Set the autocarve volume size	/cx set carvesize	51
Unit Configuration Tasks		
Create a new unit	/cx add	36
Create a hot spare	/cx add	36
Enable/disable unit write cache	/cx/ux set cache	61
Set the queue policy	/cx/ux set qpolicy	62
Set the storsave profile	/cx/ux set storsave	63
Unit Configuration Changes		
Change RAID level	/cx/ux migrate	64
Change stripe size	/cx/ux migrate	64
Expand unit capacity	/cx/ux migrate	64
Delete a unit	/cx/ux del	58
Remove a unit (export)	/cx/ux remove	58
Name a unit	/cx/ux set name	62
Controller Maintenance Tasks		
Update controller with new firmware	/cx update	39
Add a time slot to a rebuild schedule	/cx add rebuild	45

Table 4: Common Tasks Mapped to CLI Commands

Task	CLI Command	Page
Add a time slot to a verify schedule	/cx add verify	46
Add a time slot to a selftest schedule	/cx add selftest	47
Enable/disable the rebuild/migrate schedule and set the task rate	/cx set rebuild	48
Enable/disable the verify schedule and set the task rate	/cx set verify	49
Enable/disable the selftest schedule	/cx set selftest	49
View Alarms	/cx show alarms	40
Unit Maintenance Tasks		
Start a rebuild	/cx/ux start rebuild	59
Start a verify	/cx/ux start verify	59
Pause/resume rebuild	/cx/ux pause rebuild and /cx/ux resume rebuild	60
Stop verify	/cx/ux stop verify	60
Enable/disable autoverify	/cx/ux set autoverify	61
Identify all drives that make up a unit by blinking associated LEDs	/cx/ux set identify	64
Port Tasks		
Locate drive by blinking an LED	/cx/px set identify	73
Check if LED is set to on or off	/cx/px show identify	70
View information for specific drive	/cx/px show	69
View the status of specific drive	/cx/px show status	70
Enclosure Tasks		
View information about an enclosure	/ex show	79
Locate a particular drive slot in an enclosure by blinking an LED	/ex/slotx set identify	82

Syntax Overview

The command syntax uses the general form:

```
Object Command Attributes
```

Objects are shell commands, controllers, units, ports (drives), BBUs (battery backup units), and enclosures.

Commands can either select (show, get, present, read) attributes or alter (add, change, set, write) attributes.

Attributes are either Boolean Attributes or Name-Value Attributes.

- The value of a boolean attribute is deduced by presence or lack of—that is, the attribute is either specified, or not. For example, the command **show alarms** by default lists alarms with the most recent alarm first. If you include the attribute **reverse**, as in the command **show alarms reverse**, alarms are listed in reverse order.
- The value of name-value attributes are expressed in the format *attribute=value*.

Example: When adding (creating) a unit to the controller with the following command string,

```
/c1 add type=raid1 disk=0-1
```

`c1` is the object, `add` is the command, `type` (for type of array) is an attribute with `raid1` as the value of the attribute, and `disk` is another attribute with `0-1` as the value (ports 0 through 1).

Information about commands is organized by the object on which the commands act:

Shell Object Commands. Shell object commands set the focus or provide information (such as alarms, diagnostics, rebuild schedules, and so forth) about all controllers in the system. For details, see “Shell Object Commands” on page 19.

Controller Object Commands. Controller object commands provide information and perform actions related to a specific controller. For example, you use controller object commands for such tasks as seeing alarms specific to a controller, creating schedules during which background tasks are run, and setting policies for the controller. You also use the controller object command **/cx add type** to create RAID arrays. For details, see “Controller Object Commands” on page 25.

Unit Object Commands. Unit object commands provide information and perform actions related to a specific unit on a specific controller. For example, you use unit object commands for such tasks as seeing the rebuild, verify, or initialize status of a unit, starting, stopping, and resuming verifies, starting and stopping rebuilds, and setting policies for the unit. You also use the controller object command **/cx/ux migrate** to change the configuration of a RAID array. For details, see “Unit Object Commands” on page 53.

Port Object Commands. Port object commands provide information and perform actions related to a drive on a specific port. For example, you use port object commands for such tasks as seeing the status, model, or serial number of the drive. For details, see “Port Object Commands” on page 69.

BBU Object Commands. BBU object commands provide information and perform actions related to a Battery Backup Unit on a specific controller. For details, see “BBU Object Commands” on page 74.

Enclosure Object Commands. Enclosure object commands provide information and perform actions related to a particular enclosure. For example, you can use enclosure object commands to see information about an enclosure and its elements (slots, fan, and temperature sensor elements).

Help Commands. Help commands allow you to display help information for all commands and attributes. For details, see “Help Commands” on page 83.

Shell Object Commands

Shell object commands are either applicable to all the controllers in the system (such as show, rescan, flush, commit), or redirect the focused object.

Syntax

```
focus object
show [attribute [modifier]]
    ver
    alarms [reverse]
    diag
    rebuild
    verify
    selftest
rescan
flush
commit
update fw=filename_with_path [force]
```

focus Object

The focus command is active in interactive mode only and is provided to reduce typing.

The focus command will set the specified object in focus and change the prompt to reflect this. This allows you to enter a command that applies to the focus, instead of having to type the entire object name each time.

For example, where normally you might type:

```
//hostname/c0/u0 show
```

if you set the focus to `//hostname/c0/u0`, the prompt changes to reflect that, and you only have to type `show`. The concept is similar to being in a particular location in a file system and requesting a listing of the current directory.

object can have the following forms:

`//hostname/cx/ux` specifies the fully qualified URI (Universal Resource Identifier) of an object on host `hostname`, controller `cx`, unit `ux`.

`//hostname` specifies the root of host `hostname`.

`..` specifies one level up (the parent object).

`/` specifies the root at the current focused hostname.

`./object` specifies the next level of the object.

`/c0/bbu` specifies a relative path with respect to the current focused hostname.

Example:

```
//localhost> focus //elvis.amcc.com
//elvis.amcc.com>
```

```
//elvis.amcc.com> focus /c0/u0
//elvis.amcc.com/c0/u0>
```

```
//elvis.amcc.com/c0/u0> focus..
//elvis.amcc.com/c0>
```

```
//elvis.amcc.com/c0/> focus u0
//elvis.amcc.com/c0/u0>
```

```
//elvis.amcc.com/c0> focus /
//elvis.amcc.com>
```

show

This command shows a general summary of all detected controllers.

Note that the device drivers for the appropriate operating system should be loaded for the list to show all controllers. The intention is to provide a global view of the environment.

Example:

Typical output of the Show command looks like the following:

```
//localhost> show
Ctl  Model      Ports  Drives  Units  NotOpt  RRate  VRate  BBU
-----
c0   9590SE-4ME   4      4       1      0       2      5      -
```

The output above indicates that Controller 0 is a 9590SE model with 4 Ports, with 4 Drives detected (attached), total of 1 Unit, with no units in a NotOpt

(Not Optimal) state, RRate (Rebuild Rate) of 2, VRate (Verify Rate) of 5, BBU of '-' (Not Applicable). Not Optimal refers to any state except OK and VERIFYING. Other states include VERIFY-PAUSED, INITIALIZING, INIT-PAUSED, REBUILDING, REBUILD-PAUSED, DEGRADED, MIGRATING, MIGRATE-PAUSED, RECOVERY, INOPERABLE, and UNKNOWN. RRate also applies to initializing, migrating, and recovery background tasks. (Definitions of the unit statuses are available in the *3ware Serial ATA RAID Controller User Guide*.)

For a system with an enclosure unit such as the 3ware Sidecar that includes support for an EPCT (Enclosure Port Configuration Table), applicable firmware and software, and an appropriate controller (9550SX, 9590SE, or 9650SE), a global view of the environment also includes summary information about detected enclosures.

Example:

Typical output of the Show command for a system with an enclosure looks like the following:

```
//localhost> show
Ctl   Model           Ports   Drives  Units  NotOpt  RRate  VRate  BBU
-----
c0    9590SE-4ME       4       2       1      0       4      4      -

Encl  Slots   Drives  Fans   TSUnits  Ctls
-----
e0    4       2       1     1        1
```

The output above shows the enclosure summary information with the name of the enclosure, the protocol used, the number of drive slots, the number of drives, the number of fans, the number of temperature sensors, and the number of controllers that are associated with the enclosure.

show ver

This command will show the CLI and API version.

Example:

```
//localhost> show ver
CLI Version = 2.00.00.0xx
API Version = 2.00.00.0xx
```

where xx is the actual version. See the Release Notes for details.

show alarms [reverse]

This command shows the alarms or AEN messages of all controllers in the system. The default is to display the most recent message first. The **reverse** attribute displays the most recent message last.

show diag

This command shows the diagnostic information of all controllers in the system.

show rebuild

This command displays all rebuild schedules for the 9000 controllers in the system.

The rebuild rate is also applicable for initializing, migrating, and recovery background tasks.

Example:

```
//localhost> show rebuild
```

```
Rebuild Schedule for Controller /c0
```

```
=====
```

Slot	Day	Hour	Duration	Status
1	Sun	12:00am	24 hr(s)	disabled
2	Mon	12:00am	24 hr(s)	disabled
3	Tue	12:00am	24 hr(s)	disabled
4	Wed	12:00am	24 hr(s)	disabled
5	Thu	12:00am	24 hr(s)	disabled
6	Fri	12:00am	24 hr(s)	disabled
7	Sat	12:00am	24 hr(s)	disabled

```
-----
```

For additional information about rebuild schedules, see “/cx add rebuild=ddd:hh:duration” on page 45, and see the discussion of background tasks and schedules in *3ware Serial ATA RAID Controller User Guide*.

show verify

This command displays all verify schedules for the 9000 controllers in the system.

Example:

```
//localhost> show verify
```

```
Verify Schedule for Controller /c0
```

```
=====
```

Slot	Day	Hour	Duration	Status
1	Sat	11:00pm	4 hr(s)	enabled
2	-	-		enabled
3	-	-		enabled
4	-	-		enabled
5	-	-		enabled
6	-	-		enabled
7	-	-		enabled

For additional information about verify schedules, see “/cx add verify=ddd:hh:duration” on page 46, and see the discussion of background tasks and schedules in *3ware Serial ATA RAID Controller User Guide*.

show selftest

This command displays all selftest schedules for the 9000 controllers in the system.

Example:

```
//localhost> show selftest
```

```
Selftest Schedule for Controller /c0
```

```
=====
```

Slot	Day	Hour	UDMA	SMART
1	Sun	12:00am	enabled	enabled
2	Mon	12:00am	enabled	enabled
3	Tue	12:00am	enabled	enabled
4	Wed	12:00am	enabled	enabled
5	Thu	12:00am	enabled	enabled
6	Fri	12:00am	enabled	enabled
7	Sat	12:00am	enabled	enabled

For additional information about selftest schedules, see “/cx add selftest=ddd:hh” on page 47, and see the discussion of background tasks and schedules in *3ware Serial ATA RAID Controller User Guide*.

flush

This command sends a flush command to all 3ware controllers in the system. For more information, see “/cx flush” on page 39.

rescan

This command sends a rescan command to all 3ware controllers in the system. For more information, see “/cx rescan [noscan]” on page 38.

commit

This command sends a commit command to all 3ware controllers in the system. For more information, see “/cx commit” on page 39.

update fw=*filename_with_path* [force]

This command downloads the specified firmware image to the controllers that are compatible with it and iterates through all the controllers in the system, updating the firmware. For more information, see “/cx update fw=*filename_with_path* [force]” on page 39.

Controller Object Commands

Controller object commands provide information and perform actions related to a specific controller, such as **/c0**. For example, you use controller object commands to see alarms specific to a controller, to create schedules for when background tasks are run, and to set policies for the controller. You also use the controller object command **/cx add type** to create RAID arrays.

Syntax

```

/cx show
/cx show attribute [attribute ...] where attributes are:
    achip|alarms|allunitstatus|autocarve (9000 series)|
    autorebuild(9550SX and 9590SE only)|bios|
    carvesize(9000 series)|ctlbus(9550SX and 9590SE only)|
    diag|driver| drivestatus|exportjbod(9000 series)|
    firmware|memory|model|monitor|ondegrade(9500S series)
    |pcb|pchip|numdrives|numports|numunits|
    rebuild(9000 series)|selftest(9000 series)|serial|
    spinup|stagger|unitstatus|verify(9000 series)
/cx show all (where all means attributes and configurations)
/cx show diag
/cx show alarms [reverse]
/cx show rebuild (9000 series)
/cx show verify (9000 series)
/cx show selftest (9000 series)

/cx add type=<RaidType> disk=<p:-p..> [stripe=<Stripe>]
    [noscan][nocache][group=<3|4|5|6>][autoverfy][ignorECC]
    [name=string](9000 series) RaidType={raid0,raid1,raid5,
    raid10,raid50,single,spare,jbod(7000/8000 only)}
    [storsave=<protect|balance|perform>(9550SX and 9590SE
    only)]
/cx add rebuild=ddd:hh:duration (9000 only)
/cx add verify=ddd:hh:duration (9000 only)
/cx add selftest=ddd:hh (9000 only)

/cx del rebuild=slot_id (9000 only)
/cx del verify=slot_id (9000 only)
/cx del selftest=slot_id (9000 only)

/cx set exportjbod=on|off (9000 only)
/cx set ondegrade=cacheoff|follow (9500S only)
/cx set spinup=nn (9000 only)
/cx set stagger=nn (9000 only)
/cx set autocarve=on|off (9000 only)
/cx set carvesize=[1024...2048] (9000 only)
/cx set rebuild=enable|disable|1..5 (9000 only)
/cx set autorebuild=on|off (9550SX and 9590SE only)

```

```

/cx set verify=enable|disable|1..5 (9000 only)
/cx set selftest=enable|disable [task=UDMA|SMART](9000 only)

/cx flush
/cx update filename_with_path [force] (9000 only)
/cx commit (Windows only. Also known as shutdown)
/cx start mediascan (7000/8000 only)
/cx stop mediascan (7000/8000 only)
/cx rescan [noscan] (Does not import non-JBOD on 7000/8000
models.

```

/cx show

This command shows summary information on the specified controller /cx. This information is organized into a report containing two to three parts:

- A **Unit** summary listing all present units
- A **Port** summary section listing of all ports and disks attached to them.
- A **BBU** summary section listing, if a BBU is installed on the controller.
- The **Unit** summary section lists all present unit and specifies their unit number, unit type (such as RAID 5), unit status (OK, RECOVERY, INOPERABLE, UNKNOWN, DEGRADED, INITIALIZING, INIT-PAUSED, VERIFYING, VERIFY-PAUSED, REBUILDING, REBUILD-PAUSED, MIGRATING, MIGRATE-PAUSED), % Cmpl (of rebuilding, verifying, initializing, or migrating), stripe size, size (usable capacity) in gigabytes or terabytes, auto verify policy status (on/off), and ignore ECC errors policy state (on/off). (Definitions of the unit statuses are available in the *3ware Serial ATA RAID Controller User Guide*.)



Note: If an asterisk (*) appears next to the status of a unit, there is an error on one of the drives in the unit. This feature provides a diagnostic capability for potential problem drives. The error may not be a repeated error, and may be caused by an ECC error, SMART failure, or a device error. Rescanning the controller will clear the drive error status if the condition no longer exists.

The **Port** summary section lists all present ports and specifies the port number, disk status, unit affiliation, size (in gigabytes) and blocks (512 bytes), and the serial number assigned by the disk vendor.

The **BBU** summary lists details about the BBU, if one is installed. It includes a few important attributes such as hours left (in which the current BBU can backup the controller cache in the event of power loss), temperature, voltage, readiness, and so forth.

Additional attributes about controllers, units, ports and disks can be obtained by querying for them explicitly. For details, see the other show subcommands.

Typical output looks like:

```
//localhost> /c0 show
Unit  UnitType  Status  %Cmpl  Stripe  Size(GB)  Cache  AVerify  IgnECC
-----
u0    RAID-1    OK      -      -      149.05    ON     OFF      OFF
u1    RAID-5    OK      -      64k    298.22    ON     OFF      OFF
u2    SPARE     OK      -      -      149.05    ON     OFF      -

Port  Status      Unit  Size      Blocks      Serial
-----
p0    OK          u0    149.05 GB  312581808   3JS0TF14
p1    OK          u0    149.05 GB  312581808   3JS0TETZ
p2    OK          u1    149.05 GB  312581808   3JS0VG85
p3    OK          u1    149.05 GB  312581808   3JS0VGCY
p4    OK          u1    149.05 GB  312581808   3JS0VGGQ
p5    OK          u2    149.05 GB  312581808   3JS0VH1P
p6    OK          -     149.05 GB  312581808   3JS0TF0P
p7    OK          -     149.05 GB  312581808   3JS0VF43
p8    OK          -     149.05 GB  312581808   3JS0VG8D
p9    NOT-PRESENT -     -          -          -
p10   NOT-PRESENT -     -          -          -
p11   NOT-PRESENT -     -          -          -

Name  OnlineState  BBUReady  Status  Volt  Temp  Hours  LastCapTest
-----
bbu   On           Yes       OK      OK    OK    241    22-Jun-2004
```

lcx show attribute [attribute ...]

This command shows the current setting of the specified attributes on the specified controller. One or many attributes can be specified. Specifying an invalid attribute will terminate the loop. Possible attributes are: achip, allunitstatus, autocarve (9000 series), autorebuild (9550SX and 9590SE only), bios, carvesize (9000 series), driver, drivestatus, exportjbd (9000 series), firmware, memory, model, monitor, numdrives, numports, numunits, ctlbus (9550SX and 9590SE only), ondegrade (9500S), pcb, pchip, qpolicy, serial, spinup (9000 series), stagger (9000 series), and unitstatus.

Example: To see the driver and firmware installed on controller 0, enter the following:

```
//localhost> /c0 show driver firmware
/c0 Driver Version = 2.x
/c0 Firmware Version = FE9X 2.x
```

Where x=actual version

/cx show driver

This command reports the device driver version associated with controller /cx.

Example:

```
//localhost> /c0 show driver
/c0 Driver Version = 2.x
```

/cx show model

This command reports the controller model of controller /cx.

Example:

```
//localhost> /c0 show model
/c0 Model = 9500-x
```

/cx show firmware

This command reports the firmware version of controller /cx.

Example:

```
//localhost> /c0 show firmware
/c0 Firmware Version = FGXX 2.x
```

/cx show bios

This command reports the BIOS version of controller /cx.

Example:

```
//localhost> /c0 show bios
/c0 BIOS Version = BG9X 2.x
```

/cx show monitor

This command reports the monitor (firmware boot-loader) version of controller /cx.

Example:

```
//localhost> /c0 show monitor
/c0 Monitor Version = BLDR 2.x
```

/cx show serial

This command reports the serial number of the specified controller /cx.

Example:

```
//localhost> /c0 show serial  
/c0 Serial Number = F12705A3240009
```

/cx show pcb

This command reports the PCB (Printed Circuit Board) version of the specified controller /cx.

Example:

```
//localhost> /c0 show pcb  
/c0 PCB Version = RevX
```

/cx show pchip

This command reports the PCHIP (PCI Interface Chip) version of the specified controller /cx.

Example:

```
//localhost> /c0 show pchip  
/c0 PCHIP Version = 1.x
```

/cx show achip

This command reports the ACHIP (ATA Interface Chip) version of the specified controller /cx.

Example:

```
//localhost> /c0 show achip  
/c0 ACHIP Version = 3.x
```

/cx show numdrives

This command reports the number of drives currently managed by the specified controller /cx. This report does not include (logically) removed or exported drives. Also note that physically removed disk(s) will not be detected unless I/O is performed against the disk. See "/cx/px show smart" on page 71 for a workaround.

Example:

```
//localhost> /c0 show numdrives  
/c0 Number of Drives = 5
```

/cx show numports

This command reports the port capacity (number of physical ports) of the specified controller /cx.

Example:

```
//localhost> /c0 show numports
/c0 Number of Ports = 12
```

/cx show numunits

This command reports the number of units currently managed by the specified controller /cx. This report does not include off-line units (or removed units).

Example:

```
//localhost> /c0 show numunits
/c0 Number of Units = 1
```

/cx show ctlbus

This feature only applies to 9550SX and 9590SE controllers.

This command reports the controller host bus type, bus speed, and bus width.

Example:

```
//localhost> /c0 show ctlbus
/c0 Controller Bus Type = PCI-X
/c0 Controller Bus Width = 64 bits
/c0 Controller Bus Speed = 133 Mhz
```

/cx show exportjbod

This feature only applies to 9000 series controllers.

This command reports the current JBOD Export Policy: **on**, **off**, or **Not Supported**.

Example:

```
//localhost> /c0 show exportjbod
/c0 JBOD Export Policy = Not Supported.
//localhost> /c1 show exportjbod
/c1 JBOD Export Policy = on
```

/cx show spinup

This feature only applies to 9000 series controllers.

This command reports the number of concurrent disks that will spin up when the system is powered on, after waiting for the number of seconds specified with the `set stagger` command.

Example:

```
//localhost> /c0 show spinup
/c0 Disk Spinup Policy = 1
```

/cx show stagger

This feature only applies to 9000 series controllers.

This command reports the time delay between each group of spinups at the power on.

Example:

```
//localhost> /c0 show stagger
/c0 Spinup Stagger Time Policy (sec) = 2
```

/cx show ondegrade

This feature only applies to 9500S controllers.

This command reports the cache policy for degraded units. If the ondegrade policy is “Follow Unit Policy,” a unit cache policy stays the same when the unit becomes degraded. If the ondegrade policy is off, a unit cache policy will be forced to “off” when the unit becomes degraded.

Example:

```
//localhost> /c0 show ondegrade
/c0 Cache on Degraded Policy = Follow Unit Policy
```

/cx show autocarve

This feature only applies to 9000 series controllers.

This command reports the Auto-Carve policy. If the policy is **on**, all newly created or migrated units larger than the `carvesize` will be automatically carved into multiples of `carvesize` volumes plus one remainder volume. Each volume can be treated as an individual drive with its own file system. The default `carvesize` is 2TB. For more information see, “/cx show carvesize”, below.

For operating systems that support units larger than 2TB, there is no need to set the policy to **on** unless you want the operating system to have multiple smaller volumes.

If you use a 32-bit operating system, it is recommended that you keep the policy **on** unless you know that your operating system supports disks that are larger than 2 TB.

When the autocarve policy is **off**, all newly created units will consist of one single volume.

Example:

```
//localhost> /c0 show autocarve
/c0 Auto-Carving Policy = on
```

/cx show carvesize

This feature only applies to 9000 series controllers.

This command shows the maximum size of the volumes that will be created if the autocarve policy is set to **on**. The carvesize can be set between 1024 GB and 2048 GB. Default carvesize is 2048 GB (2 TB). For more information see, “/cx show autocarve” above.

Example:

```
//localhost> /c0 show carvesize
/c0 Auto-Carving Size = 2000 GB
```

/cx show memory

This command reports the size of the memory installed on the controller.



Note: The 9500S controllers ship with 128 MBytes of cache, yet only 112MB shows as memory installed. The other 16 MB is reserved for use by the controller.

Example:

```
//localhost> /c0 show memory
/c0 Memory Installed = 112MB
```

/cx show autorebuild

This feature only applies to 9550SX and 9590SE model controllers.

This command shows the Auto-Rebuild policy. If the policy is enabled, the firmware will select drives to use for rebuilding a degraded unit using the following priority order. For more information, see “/cx set autorebuild=on|off” on page 51.

1. Smallest usable spare.
2. Smallest usable unconfigured (available) drive.
3. Smallest usable failed drive.

If the policy is disabled, only spare drives will be used for an automatic rebuild operation.

Example:

```
//localhost> /c0 show autorebuild
/c0 Auto-Rebuild Policy = on
```

/cx show unitstatus

This command presents a list of units currently managed by the specified controller /cx, and shows their types, capacity, status, and unit policies.

Possible statuses include: OK, VERIFYING, VERIFY-PAUSED, INITIALIZING, INIT-PAUSED, REBUILDING, REBUILD-PAUSED, DEGRADED, MIGRATING, MIGRATE-PAUSED, RECOVERY, INOPERABLE, and UNKNOWN. (Definitions of the unit statuses are available in the *3ware Serial ATA RAID Controller User Guide*.)



Note: If an asterisk (*) appears next to the status of a unit, there is an error on one of the drives in the unit. This feature provides a diagnostic capability for potential problem drives. The error may not be a repeated error, and may be caused by an ECC error, SMART failure, or a device error. Rescanning the controller will clear the drive error status if the condition no longer exists.

Example:

```
//localhost> /c0 show unitstatus
Unit  UnitType  Status      %Cmpl  Stripe  Size(GB)  Cache  AVerify  IgnECC
-----
u0    RAID-5    OK          -      64K     223.485   OFF    OFF      ON
u1    JBOD      OK          -      -       76.3352  OFF    OFF      -
u2    RAID-0    OK          -      64K     148.99   ON     ON       -
```

/cx show allunitstatus

This command presents a count of total and Not Optimal units managed by the specified controller /cx. For more about the meaning of Not Optimal, see “Shell Object Commands” on page 19.

Example:

```
//localhost> /c0 show allunitstatus
/c0 Total Optimal Units = 2
/c0 Not Optimal Units = 0
```

/cx show drivestatus

This command reports a list of drives and their port assignment, status, the unit with which they are associated, their size in gigabytes and blocks, and the serial number assigned by the drive manufacturer. (Definitions of the drive statuses are available in the *3ware Serial ATA RAID Controller User Guide*.)

Example:

```
//localhost> /c0 show drivestatus
```

Port	Status	Unit	Size	Blocks	Serial
p0	OK	u0	149.05 GB	312581808	3JS0TF14
p1	OK	u0	149.05 GB	312581808	3JS0TETZ
p2	OK	u1	149.05 GB	312581808	3JS0VG85
p3	OK	u1	149.05 GB	312581808	3JS0VGCY
p4	OK	u1	149.05 GB	312581808	3JS0VGGQ
p5	OK	u2	149.05 GB	312581808	3JS0VH1P
p6	OK	-	149.05 GB	312581808	3JS0TF0P
p7	OK	-	149.05 GB	312581808	3JS0VF43
p8	OK	-	149.05 GB	312581808	3JS0VG8D
p9	NOT-PRESENT	-	-	-	-
p10	NOT-PRESENT	-	-	-	-
p11	NOT-PRESENT	-	-	-	-

/cx show all

This command shows the current setting of all of the following attributes on the specified controller: driver, model, firmware, memory, bios, monitor, serial, pcb, pchip, achip, numports, numunits, numdrives, unitstatus, drivestatus, allunitstatus, exportjbod, ondegrade, spinup, autocarve, and stagger.

Example: (where *x* represents the actual version number)

```
//localhost> /c0 show all
/c0 Driver Version = 3.x
/c0 Model = 9550SX-12
/c0 Memory Installed = 112MB
/c0 Firmware Version = FE9X 3.x
/c0 Bios Version = BE9X 3.x
/c0 Monitor Version = BL9X 3.x
/c0 Serial Number = xxxxx
/c0 PCB Version = Rev 0xx
/c0 PCHIP Version = 1.xx
/c0 ACHIP Version = 3.xx
/c0 Number of Ports = 12
/c0 Number of Units = 2
/c0 Number of Drives = 12
/c0 Total Optimal Units = 2
/c0 NotOptimalUnits = 0
/c0 Total Units = 2

/c0 JBOD Export Policy = off
/c0 Disk Spinup Policy = 7
/c0 Spinup Stagger Time Policy (sec) = 4
/c0 Cache on Degrade Policy = Follow Unit Policy
/c0 Auto-Carving Policy = off
/c0 Auto-Carving Size = 2047 GB
/c0 Auto-Rebuild Policy = enable
```

Unit	UnitType	Status	%Cmpl	Stripe	Size(GB)	Cache	AVerify	IgnECC
u0	RAID-5	OK	-	256K	148.99	ON	ON	ON
u1	RAID-5	OK	-	256K	595.961	ON	OFF	ON

Port	Status	Unit	Size	Blocks	Serial
p0	OK	u0	74.53 GB	156301488	3JV3MV1C
p1	OK	u0	74.53 GB	156301488	3JV3MK6B
p2	OK	u0	74.53 GB	156301488	3JV3LW52
p3	OK	u1	74.53 GB	156301488	3JV49S77
p4	OK	u1	74.53 GB	156301488	3JV3MVTA
p5	OK	u1	74.53 GB	156301488	5JV980Z0
p6	OK	u1	74.53 GB	156301488	5JV9820G
p7	OK	u1	111.79 GB	234441648	WD-WMAEL10275
p8	OK	u1	111.79 GB	234441648	WD-WMAEL10274
p9	OK	u1	111.79 GB	234441648	WD-WMAEL10281
p10	OK	u1	111.79 GB	234441648	WD-WMAEL10273
p11	OK	u1	111.79 GB	234441648	WD-WMAEL10274

```

/cx add type=<RaidType> disk=<p:-p>
[stripe=Stripe] [noscan] [group=<3/4/5/6>]
[nocache] [autoverify] [ignoreECC] [name=string]
[storsave=<protect|balance|perform>]

```

This command allows you to create a new unit on the specified controller. You specify *type*, *disks*, and optional *stripe* size. By default the host operating system will be informed of the new block device, write cache will be enabled, and a storsave policy of protect will be set. In case of RAID 50, you can also specify the layout of the unit by specifying the number of disks per disk group with the *group* attribute.

/cx is the controller name, for example /c0, /c1, and so forth.

type=RaidType specifies the type of RAID unit to be created. Possible unit types include raid0, raid1, raid5, raid10, raid50, single, spare, and JBOD (7000/8000 only).

Example: type=raid5

When a new unit is created, it is automatically assigned a unique serial number. In addition, users can assign the unit a name.



Note: The unit's serial number cannot be changed.

disk=p:-p consists of a list of ports (disks) to be used in the construction of the specified unit type. One or more ports can be specified. Multiple ports can be specified using a colon (:) or a dash (-) as port index separators. A dash indicates a range and can be mixed with colons. For example `disk=0:1:2-5:9:12` indicates port 0, 1, 2 through 5 (inclusive), 9 and 12.

stripe=Stripe consists of the stripe size to be used. The following table illustrates the supported and applicable stripes on unit types and controller models. Stripe size units are in K (kilobytes). If no stripe size is specified, 64K is used by default, if applicable. If you need to change the stripe size after the unit is created, you can do so by migrating the unit.

Table 5: Supported Stripe Sizes

Model	R0	R1	R5	R10	Spare	Single
9K	16	N/A	16	16	N/A	N/A
	64		64	64		
	256		256	256		

group=3/4/5/6 indicates the number of disks per group for a RAID 50 type. (This attribute can only be used when `type=raid50`.) Recall that a RAID 50 is a multi-tier array. At the bottom-most layer, N number of disks per group are used to form the RAID 5 layer. These RAID 5 arrays are then integrated into a RAID 0. This attribute allows you to specify the number of disks in the RAID 5 level. Valid values are 3, 4, 5 and 6.

Note that a sufficient number of disks are required for a given pattern or disk group. For example, given 6 disks, specifying 3 will create two RAID 5 arrays. With 12 disks, specifying 3 will create four RAID 5 arrays under the RAID 0 level. With only 6 disks a grouping of 6 is not allowed, as you would basically be creating a RAID 5.

The default RAID 50 grouping varies, based on number of disks. For 6 and 9 disks, default grouping is 3. For 8 disks, the default grouping is 4. For 10 disks, the default grouping is 5, and for 12 disks, the default grouping is 4. In the case of 12, the disks could be grouped into groups of 3, 4, or 6 drives. A grouping of 4 is set by default as it provides the best of net capacity and performance.

noscan attribute instructs CLI not to notify the operating system of the creation of the new unit. By default CLI will inform the operating system. One application of this feature is to prevent the operating system from creating block special devices such as `/dev/sdb` and `/dev/sdc` as some implementations might create naming fragmentation and a moving target.

nocache attribute instructs CLI to disable the write cache on the newly created unit. Enabling write cache increases write performance at the cost of potential data loss in case of sudden power loss (unless a BBU or UPS is installed). By default the cache is enabled. To avoid the possibility of data loss in the event of a sudden power loss, it is recommended not to set `nocache` unless there is a BBU (battery backup unit) or UPS (uninterruptable power supply) installed.

autoverify attribute enables the `autoverify` attribute on the unit that is to be created. For more details on this feature, see “`/cx/ux set autoverify=on|off`” on page 61. This feature is not supported on model 7000/8000. On model 9000, the JBOD `autoverify` attribute is not persistent (does not survive reboots).

ignoreECC attribute enables the `ignoreECC/OverwriteECC` attribute on the unit that is to be created. For more details on this feature, see “`/cx/ux set ignoreECC=on|off`” on page 62. The following table illustrates the supported Model-Unit Types. This table only applies to setting this feature at unit creation time. `IgnoreECC` only applies to redundant units. For the 7/8000 series, this setting is only applicable during rebuild; it is not applicable during creation.

Table 6: Supported Model-Unit Types for ignoreECC

Model	R-0	R-1	R-5	R-10	R-50	Single	JBOD	Spare
7K/8K	No	No	No	No	No	No	No	No
9K	No	Yes	Yes	Yes	Yes	No	No	No

name=string attribute allows you to name the new unit. (This feature is for 9000 series and above controllers.) The string can be up to 21 characters and cannot contain spaces. In order to use reserved characters ('<', '>', '!', '&', etc.) put double quotes (" ") around the name string. The name can be changed after the unit has been created. For more information, see “/cx/ux set name=string” on page 62 and “/cx/ux show name” on page 55.

storsave=protect|balance|perform attribute allows user to set the storsave policy of the new unit. This feature is only for 9550SX and 9590SE controllers. For more information, see “/cx/ux set storsave=protect|balance|perform [quiet]” on page 63.

/cx rescan [noscan]

This command instructs the controller to rescan all ports and reconstitute all units. The controller will update its list of ports (attached disks), and attempts to read every DCB (Disk Configuration Block) in order to re-assemble its view and awareness of logical units. Any newly found unit(s) or drive(s) will be listed.

noscan is used to not inform the operating system of the unit discovery. The default is to inform the operating system.



Note: If you are adding new drives, add them physically before issuing the rescan commands. Hot swap carriers are required unless you first power-down the system to prevent system hangs and electrical damage.

Example:

```
//localhost> /cl rescan
Rescanning controller /cl for units and drives ...Done
Found following unit(s): [/cl/u3]
Found following drive(s): [/cl/p7, /cl/p8]
```



Note: Rescanning does not import non-JBOD on 7000/8000 models.

/cx commit

This command only applies to the Windows operating system. It commits all changes if a faster shutdown method is needed when running certain database applications. Linux and FreeBSD file systems do not require this command since they have their own ways of notifying the controller to do clean up for shut down.

/cx flush

This command forces the controller to write all cached data to disk for the specified controller.

/cx update fw=*filename_with_path* [force]

This command is only for 9000 series controllers.

This command allows the downloading of the specified firmware image to the corresponding controller.

Note: Before issuing this command, you must have already obtained the firmware image and placed it on your system. You can obtain the firmware image from the 3ware website: <http://www.3ware.com/downloads>.

fw=*filename_with_path* attribute allows you to specify the firmware image file name along with its absolute path. The new image specified by this *filename_with_path* is checked for compatibility with the current controller, current driver, and current application versions. A recommendation is then made as to whether an update is needed, and you are asked to confirm whether you want to continue. If you confirm that you want to continue, the new firmware image is downloaded to the specified controller.

A reboot is required for the new firmware image to take effect.

Example:

```
//localhost> /c0 update fw=/tmp/prom0006.img
Warning: We strongly recommend backing up your data before
updating the firmware. Updating the firmware can render the
device driver and/or management tools incompatible. It is
recommended to have a copy of current firmware image for
rollbacks.
```

```
Examining compatibility data from firmware image and /c0 ...
Done.
```

New-Firmware	Current-Firmware	Current-Driver	Current-API
FE9X 3.05.00.005	FE9X 3.05.00.005	2.26.04.007	2.01.00.008

```
Current firmware version is the same as the new firmware.
```

```
Recommendation: No need to update.
```

```
Given the above recommendation...
Do you want to continue ? Y|N [N]: y
Downloading the firmware from file /tmp/prom0006.img ... Done.
The new image will take effect after reboot.
```

force attribute is optional. If you include it, the compatibility checks are bypassed.

/cx show alarms [reverse]

Asynchronous events (also referred to as AENs or alarms) are originated by firmware and captured by their respective device drivers. These events reflect warning, debugging, and/or informative messages for the end user. These events are kept in a finite queue inside the kernel, awaiting extraction by user space programs such as CLI and/or 3DM.

The `/cx show alarms` command displays all available alarms on a given controller. The default is to display the most recent alarm or AEN message first. The user can also use the `[reverse]` attribute to display the most recent alarm or AEN message last.

Alarms generated on 7000/8000 controllers do not have dates, so you will see a '-' in the Date column. This means that it is not applicable. In addition, alarm messages on 7000/8000 controllers contain the severity in the message text, so the Severity column also shows a '-'.

Typical output looks like:

```
tw_cli> /cl show alarms reverse
Ctl Date                               Severity Message
-----
c1 [Fri Nov 28 04:26:31 2003] ERROR (0x04:0x0002): Degraded unit detected:unit=0, port=2
c1 [Fri Nov 28 06:13:54 2003] INFO (0x04:0x000B): Rebuild started: unit=0
c1 [Fri Nov 28 06:30:35 2003] INFO (0x04:0x003B): Background rebuild paused:unit=0
c1 [Fri Nov 28 06:33:00 2003] ERROR (0x04:0x0002): Degraded unit detected:unit=0, port=0
c1 [Fri Nov 28 06:33:04 2003] ERROR (0x04:0x0002): Degraded unit detected:unit=0, port=4
c1 [Fri Nov 28 06:33:46 2003] INFO (0x04:0x000B): Rebuild started: unit=0
c1 [Fri Nov 28 06:37:58 2003] INFO (0x04:0x000B): Rebuild started: unit=0
c1 [Fri Nov 28 07:51:34 2003] INFO (0x04:0x0005): Background rebuild done:unit=0
c1 [Fri Nov 28 07:59:43 2003] INFO (0x04:0x0005): Background rebuild done:unit=0
c1 [Mon Dec 1 02:26:12 2003] ERROR (0x04:0x0002): Degraded unit detected:unit=0, port=3
```

/cx show diag

This command extracts controller diagnostics suitable for technical support usage. Note that some characters might not be printable or rendered correctly (human readable). It is recommended to save the output from this command to a file, where it can be communicated to technical support or further studied with Linux utilities such as `od(1)`.

In order to redirect the output you must run the following command from a command line, not from within the `tw_cli` shell.

```
tw_cli /c0 show diag > diag.txt
```

/cx show rebuild

9000 series controllers support background tasks and allow you to schedule a regular time when they occur.

Rebuild is one of the supported background tasks. Migrate and initialize are other background tasks that follow the same schedule as rebuild. Other background tasks for which there are separate schedules are verify and selftest. For each background task, up to 7 time periods can be registered, known as slots 1 through 7. Each task schedule can be managed by a set of commands including **add**, **del**, **show** and **set** a task. Background task schedules have a slot id, start-day-time, duration, and status attributes.

For details about setting up a schedule for background rebuild tasks, see “Setting Up a Rebuild Schedule” on page 45.

Rebuild activity attempts to (re)synchronize all members of redundant units such as RAID-1, RAID-10, RAID-5 and RAID-50. Rebuild can be started manually or automatically if a spare has been defined. Scheduled rebuilds will take place during the scheduled time slot, if enabled the schedules are enabled. For in depth information about rebuild and other background tasks, see “About Background Tasks” in the *3ware Serial ATA RAID Controller User Guide*.

The `show rebuild` command displays the current rebuild background task schedule as illustrated below.

```
//localhost> /c1 show rebuild
Rebuild Schedule for Controller /c1
=====
Slot      Day      Hour      Duration      Status
-----
1         Mon      2:00pm    10 hr(s)      disabled
2         Thu      7:00pm    18 hr(s)      disabled
3         -        -         -              disabled
4         -        -         -              disabled
5         -        -         -              disabled
6         Mon      1:00am    4 hr(s)       disabled
7         Sun      12:00am   1 hr(s)       disabled
```

A status of “disabled” indicates that the task schedule is disabled. In this case, the controller will not use the defined schedule timeslots. If the rebuild command is entered manually, rebuilding will start within 10 to 15 minutes. It will begin automatically if a rebuild is needed and a proper spare drive is set up.

If the rebuild schedule is enabled while a rebuild process is underway, the rebuild will pause until a scheduled time slot.

Example:

If a unit is in the initialization state at noon on Wednesday and the rebuild schedule shown above is in use (with schedules disabled), you would see the following status using the show command:

```
//localhost>/cl show
Unit  UnitType  Status      %Cmpl  Stripe  Size(GB)  Cache  AVerify  IgnECC
-----
u0    RAID-5    INITIALIZING  0      64K     521.466   ON     OFF     OFF

Port  Status      Unit  Size      Blocks      Serial
-----
p0    NOT-PRESENT  -    -          -            -
p1    OK           u0    76.33 GB  160086528   Y2NXL7FE
p2    NOT-PRESENT  -    -          -            -
p3    OK           u0    76.33 GB  160086528   Y2NXLB9E
p4    NOT-PRESENT  -    -          -            -
p5    OK           u0    76.33 GB  160086528   Y2NXQPZE
p6    NOT-PRESENT  -    -          -            -
p7    OK           u0    76.33 GB  160086528   Y2NXM4VE
p8    OK           u0    74.53 GB  156301488   3JV3WTSE
p9    OK           u0    74.53 GB  156301488   3JV3WRHC
p10   OK           u0    74.53 GB  156301488   3JV3WQLQ
p11   OK           u0    74.53 GB  156301488   3JV3WQLZ
```

If you then enable the rebuild schedules, the unit initialization will be paused until the next scheduled time slot, as reflected in the examples below:

```
//localhost> /cl set rebuild=enable
Enabling scheduled rebuilds on controller /cl ...Done.

//localhost> /cl show rebuild
Rebuild Schedule for Controller /cl
=====
Slot  Day    Hour      Duration    Status
-----
1     Mon    2:00pm    10 hr(s)    enabled
2     Thu    7:00pm    18 hr(s)    enabled
3     -      -         -           -
4     -      -         -           -
5     -      -         -           -
6     Mon    1:00am    4 hr(s)     enabled
7     Sun    12:00am   1 hr(s)     enabled

//localhost> /cl show
Unit  UnitType  Status      %Cmpl  Stripe  Size(GB)  Cache  AVerify  IgnECC
-----
u0    RAID-5    INIT-PAUSED  0      64K     521.466   ON     OFF     OFF

Port  Status      Unit  Size      Blocks      Serial
-----
p0    NOT-PRESENT  -    -          -            -
p1    OK           u0    76.33 GB  160086528   Y2NXL7FE
p2    NOT-PRESENT  -    -          -            -
p3    OK           u0    76.33 GB  160086528   Y2NXLB9E
p4    NOT-PRESENT  -    -          -            -
p5    OK           u0    76.33 GB  160086528   Y2NXQPZE
p6    NOT-PRESENT  -    -          -            -
p7    OK           u0    76.33 GB  160086528   Y2NXM4VE
p8    OK           u0    74.53 GB  156301488   3JV3WTSE
```

```

p9      OK          u0      74.53 GB   156301488  3JV3WRHC
p10     OK          u0      74.53 GB   156301488  3JV3WQLQ
p11     OK          u0      74.53 GB   156301488  3JV3WQLZ

```

/cx show verify

9000 series controllers support background tasks and allow you to schedule a regular time when they occur.

Verify is one of the supported background tasks. Rebuild and selftest are other background tasks for which there are separate schedules. Migrate and initialize are additional background tasks that follow the same schedule as rebuild. For each background task, up to 7 time periods can be registered, known as slots 1 through 7. Each task schedule can be managed by a set of commands including **add**, **del**, **show** and **set** a task. Background task schedules have a slot id, start-day-time, duration, and status attributes.

For details about setting up a schedule for background verify tasks, see “Setting Up a Verify Schedule” on page 46.

Verify activity verifies all units based on their unit type. Verifying RAID-1 involves checking that both drives contain the exact data. On RAID-5, the parity information is used to verify data integrity. RAID-10 and 50 are composite types and follow their respective array types. On 9000 series, non-redundant units such as RAID-0, JBOD, single, and spare, are also verified (by reading and reporting un-readable sectors). If any parity mismatches are found, the array will be automatically background initialized. (For information about the initialization process, see the user guide that came with your 3ware RAID controller.)

The `show verify` command displays the current verify background task schedule as illustrated below.

```

//localhost> /c1 show verify
Verify Schedule for Controller /c1
=====
Slot    Day      Hour          Duration      Status
-----
1       Mon      2:00am        4 hr(s)       disabled
2       -        -             -             disabled
3       Tue      12:00am       24 hr(s)      disabled
4       Wed      12:00am       24 hr(s)      disabled
5       Thu      12:00am       24 hr(s)      disabled
6       Fri      12:00am       24 hr(s)      disabled
7       Sat      12:00am       24 hr(s)      disabled

```

A status of “disabled” indicates that the controller will not use the defined schedule timeslots and will start verifying immediately (within 10 to 15 minutes), if the verify command is entered manually, or it will begin automatically if the autoverify option is set. Rebuilds, migrations, and initializations will take priority over verifies.

/cx show selftest

9000 series controllers support background tasks and allow you to schedule a regular time when they occur.

Selftest is one of the supported background tasks. Rebuild and verify are other background tasks for which there are separate schedules. Migrate and initialize are additional background tasks that follow the same schedule as rebuild. For each background task, up to 7 time periods can be registered, known as slots 1 through 7. Each task schedule can be managed by a set of commands including **add**, **del**, show and **set** a task. Background task schedules have a slot id, start-day-time, duration, and status attributes.

For details about setting up a schedule for background selftest tasks, see “Setting Up a Selftest Schedule” on page 47.

Selftest activity provides two types of selftests; UDMA (Ultra Direct Memory Access) and SMART (Self Monitoring Analysis and Reporting). Both self tests are checked once each day by default.

UDMA self test entails checking the current ATA bus speed (between controller and attached disk), which could have been throttled down during previous operations and increase the speed for best performance (usually one level higher). Possible speeds include 33, 66, 100 and 133 Mhz (at this writing). Note that UDMA selftest is not applicable (or required) with SATA drives, but is left enabled by default.

SMART activity instructs the controller to check certain SMART supported thresholds by the disk vendor. An AEN is logged to the alarms page if a drive reports a SMART failure.

The **show selftest** command displays the current selftest background task schedule as illustrated below. Selftests do not have a time duration since they are completed momentarily.

```
//localhost> /cl show selftest
```

```
Selftest Schedule for Controller /c1
```

```
=====
```

Slot	Day	Hour	UDMA	SMART
1	Sun	12:00am	enabled	enabled
2	Mon	12:00am	enabled	enabled
3	Tue	12:00am	enabled	enabled
4	Wed	12:00am	enabled	enabled
5	Thu	12:00am	enabled	enabled
6	Fri	12:00am	enabled	enabled
7	Sat	12:00am	enabled	enabled

```
-----
```

`/cx add rebuild=ddd:hh:duration`

This command adds a new background rebuild task to be executed on the day *ddd* (where *ddd* is Sun, Mon, Tue, Wed, Thu, Fri, and Sat), at the hour *hh* (range 0 .. 23), for a duration of *duration* (range 1 .. 24) hours. A maximum of seven rebuild tasks can be scheduled. This command will fail if no (empty) task slot is available.

Example:

```
//localhost> /c1 add rebuild=Sun:16:3
```

adds a rebuild background task schedule to be executed on Sundays at 16 hours (4:00 PM) for a duration of 3 hours.

Setting Up a Rebuild Schedule

Setting up a rebuild schedule requires several steps, and several different CLI commands in addition to `/cx add rebuild`.

To set up the rebuild schedule you want to use, follow this process:

- 1 Use the `/cx show rebuild` command to display the current schedule for rebuild tasks. (For details, see page 41.)
- 2 If any of the scheduled tasks do not match your desired schedule, use the `/cx del rebuild` command to remove them. (For details, see page 47.)
- 3 Use the `/cx add rebuild` command to create the rebuild schedule slots you want (described above.)
- 4 Use the `/cx set rebuild=enable` command to enable the schedule (this enables all rebuild schedule slots). (For details, see page 48.)



Warning: If all time slots are removed, be sure to also disable the schedule. Otherwise the applicable background task will never occur.

`/cx add verify=ddd:hh:duration`

This command adds a new background verify task to be executed on the day *ddd* (where *ddd* is Sun, Mon, Tue, Wed, Thu, Fri, and Sat), at hour *hh* (range 0 .. 23), for a duration of *duration* (range 1 .. 24) hours. A maximum of seven verify tasks can be scheduled. This command will fail if no (empty) task slot is available.

Example:

```
//localhost> /cl add verify=Sun:16:3
```

adds a verify background task schedule to be executed on Sundays at 16 hours (4:00 PM) for a duration of 3 hours.

Setting Up a Verify Schedule

Setting up a verify schedule requires several steps, and several different CLI commands in addition to `/cx add verify`.

To set up the verify schedule you want to use, follow this process:

- 1 Use the `/cx show verify` command to display the current schedule for verify tasks. (For details, see page 43.)
- 2 If any of the scheduled tasks do not match your desired schedule, use the `/cx del verify` command to remove them. (For details, see page 48.)
- 3 Use the `/cx add verify` command to create the verify schedule slots you want (described above.)
- 4 Use the `/cx set verify=enable` command to enable the schedule (this enables all rebuild schedule slots). (For details, see page 49.)
- 5 Use the `/cx/ux set autoverify=on` command to turn on autoverify for each unit you want to follow the schedule. (For details, see page 61.)



Note: If you do not enable autoverify for units or start a verification manually, your verify schedule will not run, even if it is enabled with the `/cx set verify=enable` command.



Warning: If all time slots are removed, be sure to also disable the schedule. Otherwise the applicable background task will never occur.

`/cx add selftest=ddd:hh`

This command adds a new background selftest task to be executed on the day *ddd* (where *ddd* is Sun, Mon, Tue, Wed, Thu, Fri, and Sat), at hour *hh* (range 0 .. 23). Notice that selftest runs to completion and as such no duration is provided. A maximum of seven selftest tasks can be scheduled. This command will fail if no (empty) task slot is available.

Example:

```
//localhost> /c1 add selftest=Sun:16
```

adds a selftest background task schedule to be executed on Sundays at 16 hours (4:00 PM).

Setting Up a Selftest Schedule

Setting up a selftest schedule requires several steps, and several different CLI commands in addition to `/cx add selftest`.

To set up the selftest schedule you want to use, follow this process:

- 1 Use the `/cx show selftest` command to display the current schedule for selftest tasks. (For details, see page 44.)
- 2 If any of the scheduled tasks do not match your desired schedule, use the `/cx del selftest` command to remove them. (For details, see page 48.)
- 3 Use the `/cx add selftest` command to create the selftest schedule slots you want (described above.)
- 4 Use the `/cx set selftest=enable` command to enable the schedule (this enables all selftest schedule slots). (For details, see page 49.)

`/cx del rebuild=slot_id`

This command removes the rebuild background task in slot *slot_id*.

Example:

```
//localhost> /c1 del rebuild=2
```

removes the rebuild background task in slot 2.



Warning: If all time slots are removed, be sure to also disable the schedule. Otherwise the applicable background task will never occur

`/cx del verify=slot_id`

This command removes the verify background task in slot *slot_id*.

Example:

```
//localhost> /cl del verify=3
```

removes verify background task in slot 3.



Warning: If all time slots are removed, be sure to also disable the schedule. Otherwise the applicable background task will never occur

`/cx del selftest=slot_id`

This command removes (or unregisters) the selftest background task in slot *slot_id*.

Example:

```
//localhost> /cl del selftest=3
```

Will remove selftest background task in slot 3.



Warning: If all time slots are removed, be sure to also disable the schedule. Otherwise the selftest background task will never occur.

`/cx set rebuild=enable|disable|1..5`

This command enables or disables all rebuild background task slots on controller */cx* and sets the priority of rebuild versus I/O operations. When enabled, rebuild tasks will only be run during the time slots scheduled for rebuilds. If a rebuild is taking place when the schedule is enabled, it will be paused until the next scheduled time.

The priority of rebuild versus I/O operations is specified with *1..5*, where 1 is more resources and 5 the least. Setting the value to 1 gives maximum processing time to rebuilds rather than I/O. Setting the value to 5 gives maximum processing time to I/O rather than rebuilds.

Enabling and disabling rebuild schedules is only for 9000 models, however the rebuild rate (**1..5**) applies to all controllers.

7000- and 8000-series controllers have only one setting for Task Rate; it applies to both rebuild and verify rates. This rate is not persistent following a reboot for 7000- and 8000-series controllers.

/cx set verify=enable|disable|1..5

This command enables or disables all verify background task slots on controller /cx and (when enabled) sets the priority of verification versus I/O operations. When enabled, verify tasks will only be run during the time slots scheduled for verifies. If a verify is taking place when the schedule is enabled, it will be paused until the next scheduled time.

The priority of verify versus I/O operations is specified with 1..5, where 1 is more resources and 5 the least. Setting this value to 1 implies fastest verify, and 5 implies fastest I/O.

Enabling and disabling verify schedules is only for 9000 models, however the verify rate (1..5) applies to all controllers.



Note: When enabling the verify schedule you must also remember to enable the autoverify setting for the units to be verified. For more information see “/cx/ux set autoverify=on|off” on page 61.

/cx set selftest=enable|disable [task=UDMA|SMART]

This command enables or disables all selftest tasks or a particular selftest_task (UDMA or SMART).

Enabling and disabling selftest is only for 9000 models. 7/8000 models have the same internal schedule, but it is not viewable or changeable.

Example:

```
//localhost> /c0 selftest=enable task=UDMA
```

enables UDMA selftest on controller c0.

/cx set exportjbod=on|off

This command allows you to set the JBOD Export Policy to **on** or **off**. By default, exportjbod is **off**.

If the JBOD export policy is **off**, CLI will not be able to create JBODs. During reboot, firmware will not export JBOD units to the operating system.

The JBOD Export Policy is only supported on 9000-series controllers. Previous models did not have such a policy enforcement feature.

A JBOD is an unconfigured disk attached to your 3ware RAID controller. AMCC recommends that you use Single Disk as a replacement for JBOD, to take advantage of features such as RAID level migration.

`/cx set ondegrade=cacheoff|follow`

This command is only for 9500S controllers.

This command allows you to set a controller-based write cache policy. If the policy is set to **cacheoff** and a unit degrades, the firmware will disable the write-cache on the degraded unit, regardless of what the unit-based write cache policy is. If the policy is set to **follow** and a unit degrades, firmware will follow whatever cache policy has been set for that unit. (For details about the unit-based policy, see “/cx/ux set cache=on|off [quiet]” on page 61.)

`/cx set spinup=nn`

This command is only for 9000 series controllers.

This command allows you to set a controller-based Disk Spinup Policy that specifies how many drives can spin up at one time. The value must be a positive integer between 1 and the number of disks/ports supported on the controller (4, 8, or 12). The default is 1.

This policy is used to stagger spinups of disks at boot time in order to spread the power consumption on the power supply. For example, given a spinup policy of 2, the controller will spin up two disks at a time, pause, and then spin up another 2 disks. The amount of time to pause can be specified with the Spinup Stagger Time Policy (`/cx set stagger`).

Not all drives support staggered spinup. If you enable staggered spinup and have drives that do not support it, the setting will be ignored.

`/cx set stagger=nn`

This command is only for 9000 series controllers.

This command allows you to set a controller-based Disk Spinup Stagger Time Policy that specifies the delay between spin-ups. The value must be a positive integer between 0 to 60 seconds. This policy, in conjunction with Disk Spinup Policy, specifies how the controller should spin up disks at boot time. The default is 6 seconds.

`/cx set autocarve=on|off`

This command is only for 9000 series controllers.

This command allows you to set the auto-carve policy to **on** or **off**. By default, autocarve is **off**.

When the auto-carve policy is set to **on**, any unit larger than the carvesize is created or migrated into one or more carvesize volumes and a remaining

volume. Each volume can then be treated as an individual disk with its own file system. The default carvesize is 2 TB.

This feature is useful for operating systems limited to 2TB file systems.

For example, using the 2 TB default carvesize, a 3 TB unit will be configured into one 2 TB volume and one 1 TB volume. A 5 TB unit will be configured into two 2 TB volumes and one 1 TB volume.

When auto-carve policy is set to **off**, all new units are created as a single large volume. If the operating system can only recognize up to 2 TBs, space over 2 TB will not be available.

Example:

```
//localhost> /c0 set autocarve=on
Setting Auto-Carving Policy on /c0 to on ... Done.
```

/cx set carvesize=[1024..2048]

This command is only for 9000 series controllers.

This command allows you to set the carve size in GB. This feature works together with autocarve. See “/cx set autocarve=on|off” above for details.

Example:

```
//localhost> /c0 set carvesize=2000
Setting Auto-Carving Size on /c0 to 2000 GB ... Done.
```

/cx set autorebuild=on|off

This command is only for 9550SX and 9590SE controllers.

This command turns the Auto-Rebuild policy on or off. By default, autorebuild is on.

If the policy is on the firmware will select drives to use for rebuilding a degraded unit using the following priority order.

1. Smallest usable spare.
2. Smallest usable unconfigured (available) drive.
3. Smallest usable failed drive.



Note: Failed drives can be drives that have mechanically failed, or they can be drives that have been disconnected from the controller long enough for the controller to classify them as failed.

Enabling Auto-Rebuild allows you to add a drive to the controller and have it be available for a rebuild as soon as you tell the controller to rescan, without

having to specify it as a spare. It also means that if you accidentally disconnect a drive (causing the controller to see it as a failed drive) and then reconnect it, the controller will automatically try to use it again.

If the policy is off, spares are the only candidates for rebuild operations.

Example:

```
//localhost> /c0 set autorebuild=enable  
Setting Auto-Rebuild Policy on /c0 to enable ... Done.
```

/cx start mediascan

This command applies only to 7000/8000 controllers. For 9000 series controllers, use the verify command.

This command provides media scrubbing for validating the functionality of a disk, including bad block detection, remapping, and so forth. The command starts a media scan operation on the specified controller */cx*.

/cx stop mediascan

This command applies only to 7000/8000 controllers.

This command stops a media scan operation on the specified controller */cx*. (Media scans are started using **/cx start mediascan**.)

Unit Object Commands

Unit Object commands provide information and perform actions related to a specific unit, such as `/c0/u1` (unit 1 on controller 0). For example, you use logical disk object commands for such tasks as seeing the rebuild, verify, or initialize status of a unit, starting, stopping, and resuming rebuilds and verifies, and setting policies for the unit.

Syntax

```

/cx/ux show
/cx/ux show attribute [attribute ...] where attributes are:
    initializestatus|name(9000 series)|
    qpolicy(9550SX and 9590SE only)|rebuildstatus|
    serial(9000 series)|status|verifystatus|
    storsave(9550SX and 9590SE only)|volumes(9000 series)|
    identify (9550SX and 9590SE only)
/cx/ux show all

/cx/ux start rebuild disk=<p:-p...> [ignoreECC]
/cx/ux start verify
/cx/ux pause rebuild           (7000/8000 only)
/cx/ux resume rebuild         (7000/8000 only)
/cx/ux stop verify
/cx/ux flush
/cx/ux del [noscan] [quiet]
/cx/ux set autoverify=on|off
/cx/ux set cache=on|off [quiet]
/cx/ux set identify=on|off     (9550SX and 9590SE only)
/cx/ux set ignoreECC=on|off
/cx/ux set qpolicy=on|off     (9550SX and 9590SE only)
/cx/ux set name=string         (9000 series)
/cx/ux set storsave=protect|balance|perform [quiet](9550SX
    and 9590SE only)
/cx/ux migrate type=RaidType [disk=p:-p] [group=3|4|5|6]
    [stripe=Stripe] [noscan] [nocache] [autoverify]
    (9000 only)
/cx/ux remove [noscan] [quiet]

```

`/cx/ux show`

This command shows summary information about the specified unit `/cx/ux`. If the unit consists of sub-units as with the case of RAID-1, RAID-5, RAID-10, RAID-50, then each sub-unit is further presented. If the Auto-Carving policy was on at the time the unit was created and the unit is over the carve size, multiple volumes were created and are displayed at the end of the summary information.

One application of the `/cx/ux show` command is to see which sub-unit of a degraded unit has caused the unit to degrade and which disk within that sub-unit is the source of degradation. Another application is to see the source and destination units during a migration.

Example:

```
//localhost> /c0/u1 show
Unit      UnitType  Status      %Cmpl  Port  Stripe  Size(GB)  Blocks
-----
u1        RAID-0    OK          -      -     64K      3576.06   7499550720
u1-0      DISK      OK          -      p0    -        298.01   624962560
u1-1      DISK      OK          -      p1    -        298.01   624962560
u1-2      DISK      OK          -      p2    -        298.01   624962560
u1-3      DISK      OK          -      p3    -        298.01   624962560
u1-4      DISK      OK          -      p4    -        298.01   624962560
u1-5      DISK      OK          -      p5    -        298.01   624962560
u1-6      DISK      OK          -      p6    -        298.01   624962560
u1-7      DISK      OK          -      p7    -        298.01   624962560
u1-8      DISK      OK          -      p8    -        298.01   624962560
u1-9      DISK      OK          -      p9    -        298.01   624962560
u1-10     DISK      OK          -      p10   -        298.01   624962560
u1-11     DISK      OK          -      p11   -        298.01   624962560
u1/v0     Volume   -          -      -     -        2047.00  4292870144
u1/v1     Volume   -          -      -     -        1529.06  3206680576
```

`/cx/ux show attribute [attribute ...]`

This command shows the current setting of the specified attributes. One or many attributes can be requested. Specifying an invalid attribute will terminate the loop. Possible attributes are: `initializestatus`, `name` (9000 series), `qpolicy` (9550SX and 9590SE only), `rebuildstatus`, `serial` (9000 series), `status`, `storsave` (9550SX and 9590SE only), `verifystatus`, `volumes` (9000 series), `autoverify` (9000 series), `cache`, `ignoreECC` (9000 series), and `identify` (9550SX and 9590SE only).

`/cx/ux show status`

This command reports the status of the specified unit.

Possible statuses include: OK, VERIFYING, VERIFY-PAUSED, INITIALIZING, INIT-PAUSED, REBUILDING, REBUILD-PAUSED, DEGRADED, MIGRATING, MIGRATE-PAUSED, RECOVERY, INOPERABLE, and UNKNOWN. (Definitions of the unit statuses are available in the *3ware Serial ATA RAID Controller User Guide*.)

Example:

```
//localhost> /c0/u0 show status
/c0/u5 status = OK
```

/cx/ux show rebuildstatus

This command reports the rebuildstatus (if any) of the specified unit.

Example:

```
//localhost> /c0/u5 show rebuildstatus  
/c0/u5 is not rebuilding, its current state is OK
```

If the unit is in the process of migrating, the command will return the following:

```
//localhost> /c0/u5 show rebuildstatus  
/c0/u5 is not rebuilding, its current state is MIGRATING
```

/cx/ux show verifystatus

This command reports the verifystatus (if any) of the specified unit.

Example:

```
//localhost> /c0/u5 show verifystatus  
/c0/u5 is not verifying, its current state is OK
```

/cx/ux show initializestatus

This command reports the initializestatus (if any) of the specified unit.

Example:

```
//localhost> /c0/u5 show initializestatus  
/c0/u5 is not initializing, its current state is OK
```

/cx/ux show name

This feature only applies to 9000 series controllers.

This command reports the name (if any) of the specified unit.

Example:

```
//localhost> /c0/u5 show name  
/c0/u5 name = Joe
```

/cx/ux show serial

This feature only applies to 9000 series controllers.

This command reports the unique serial number of the specified unit.

Example:

```
//localhost> /c0/u5 show serial  
/c0/u5 Serial Number = 12345678901234567890
```

/cx/ux show qpolicy

This feature only applies to 9550SX and 9590SE model controllers.

This command reports the queue policy of the firmware. If the queue policy is on, the firmware utilizes the drive queueing policy. If some drives in the unit do not support a queueing policy, this policy will have no effect on those drives.

Note that currently only NCQ will be enabled, not tag-queueing.

Example:

```
//localhost> /c0/u5 show qpolicy
/c0/u5 Command Queuing Policy = on
```

/cx/ux show storsave

This feature only applies to 9550SX and 9590SE model controllers.

This command reports the storsave policy on the unit.

For more information see, “/cx/ux set storsave=protect|balance|perform [quiet]” on page 63.

Example:

```
//localhost> /c0/u5 show storsave
/c0/u5 Command Storsave Policy = protect
```

/cx/ux show identify

This feature only applies to 9550SX and 9590SE model controllers.

This command is related to the **/cx/ux set identify** command. It shows the identify status of the specified unit (either on or off).

Example:

```
//localhost> /c0/u0 show identify
/c0/u0 Identify status = on
```

/cx/ux show autoverify

This feature only applies to 9000 series controllers.

This command shows the current autoverify setting of the specified unit.

Example:

```
//localhost> /c0/u0 show autoverify
/c0/u0 Auto Verify Policy = off
```

/cx/ux show cache

This command shows the current write cache state of the specified unit.

Example:

```
//localhost> /c0/u0 show cache
/c0/u0 Cache State = on
```

/cx/ux show ignoreECC

This feature only applies to 9000 series controllers.

This command shows the current setting of the ignoreECC policy for the specified unit.

Example:

```
//localhost> /c0/u0 show ignoreECC
/c0/u0 Ignore ECC policy = off
```

/cx/ux show volumes

This feature only applies to 9000 series controllers.

This command reports the number of volumes in the specified unit. The number of volumes will normally be “1” unless the drive capacity exceeds 2TB and auto-carving is enabled.

/cx/ux show all

This command shows the current setting of all above attributes.

If the auto-carve policy was on at the time the unit was created and the unit is over the carve size, multiple volumes were created and are displayed at the end of the summary information.

Example:

```
//localhost> /c0/u1 show all
/c0/u1 status = OK
/c0/u1 is not rebuilding, its current state is OK
/c0/u1 is not verifying, its current state is OK
/c0/u1 is not initializing, its current state is OK
/c0/u1 volume(s) = 2
/c0/u1 name = 1234567
/c0/u1 serial number = C6CPR7JMF98DA8001DF0
```

Unit	UnitType	Status	%RCmpl	%VlmCmpl	Port	Stripe	Size(GB)
u1	RAID-0	OK	-	-	-	64K	3578.40
u1-0	DISK	OK	-	-	p0	-	298.20
u1-1	DISK	OK	-	-	p1	-	298.20

u1-2	DISK	OK	-	-	p2	-	298.20
u1-3	DISK	OK	-	-	p3	-	298.20
u1-4	DISK	OK	-	-	p4	-	298.20
u1-5	DISK	OK	-	-	p5	-	298.20
u1-6	DISK	OK	-	-	p6	-	298.20
u1-7	DISK	OK	-	-	p7	-	298.20
u1-8	DISK	OK	-	-	p8	-	298.20
u1-9	DISK	OK	-	-	p9	-	298.20
u1-10	DISK	OK	-	-	p10	-	298.20
u1-11	DISK	OK	-	-	p11	-	298.20
u1/v0	VOLUME	-	-	-	-	-	500.00
u1/v1	VOLUME	-	-	-	-	-	3078.40

/cx/ux remove [noscan] [quiet]

This command allows you to remove (or export) a unit. Exporting a unit instructs the firmware to remove the specified unit from its poll of managed units, but retains the DCB (Disk Configuration Block) metadata. As such the unit can later be re-imported.

noscan is used to not inform the operating system of this change. The default is to inform the operating system.

quiet is used for non-interactive mode. No confirmation is given and the command is executed immediately. This is useful for scripting purposes.

Example of interactive mode:

```
//localhost> /c0/u0 export
Are you sure you want to export or remove the unit /c0/u0? Enter
Y to continue.
```



Note: After the unit is removed through the CLI, the unit can be physically removed. Hot swap carriers are required to do this while the system is online. Otherwise you must power down the system to prevent system hangs and damage.

/cx/ux del [noscan] [quiet]

This command allows you to delete a unit. Deleting a unit not only removes the specified unit from the controller's list of managed units, but also destroys the DCB (Disk Configuration Block) metadata. After deleting a unit, ports (or disks) associated with the unit will be part of the free pool of managed disks.



Warning: This is a destructive command and should be used with care. All data on the specified unit will be lost after executing this command.

noscan is used to not inform the operating system of this change. The default is to inform the operating system.

quiet is used for non-interactive mode. No confirmation is given and the command is executed immediately. This is useful for scripting purposes.

Example of interactive mode:

```
//localhost> /c0/u0 del
Are you sure you want to delete the unit /c0/u0? Enter Y to
continue.
```

`/cx/ux start rebuild disk=p<p:-p...> [ignoreECC]`

This command allows you to rebuild a degraded unit using the specified **disk=*p***. Rebuild only applies to redundant arrays such as RAID 1, RAID 5, RAID 10, and RAID 50.

During rebuild, bad sectors on the source disk will cause the rebuild to fail. You can allow the operation to continue by using *ignoreECC*.

The rebuild process is a background task and will change the state of a unit to REBUILDING. Various show commands also show the percent completion as rebuilding progresses.

Note that the disk used to rebuild a unit (specified with **disk=*p***) must be a SPARE or a unconfigured disk. You must first remove the degraded drive(s) before starting the rebuild. Refer to the command “`/cx/px remove [noscan] [quiet]`” on page 73 for details. Also refer to the command “`/cx rescan [noscan]`” on page 38 to add new drives or to retry the original drive.

If you are rebuilding a RAID 50 or RAID 10 unit, multiple drives can be specified if more than one sub-array is degraded.

When you issue this command, the specified rebuild will begin if schedules are disabled; otherwise it will pause until the next scheduled rebuild. A file system check is recommended following rebuild when using the *ignoreECC* option.

`/cx/ux start verify`

This command starts a background verification process on the specified unit **/cx/ux**. The following table shows the supported matrix as a function of the controller model and logical unit type.

N/A (Not Applicable) refers to cases where the given logical unit type is not supported on that controller model.

Table 7: Supported RAID (Logical Unit) Types for Verification

Model	R0	R1	R5	R10	Single	Spare
7K/8K	No	Yes	Yes	Yes	N/A	No
9K	Yes	Yes	Yes	Yes	Yes	Yes

When you issue this command, the specified verify will begin if schedules are disabled; otherwise it will pause until the next scheduled verify. Verify will also pause if a rebuild or initialization is currently in progress.

`/cx/ux pause rebuild`

This command allows you to pause the rebuild operation on the specified unit `/cx/ux`.

This feature is only supported on the 7000/8000 series controllers. 9000 series controllers have an on-board scheduler where rebuild operations can be scheduled to take place at specified start and stop times. The `/cx/ux pause rebuild` command is provided to enable 7000/8000 users to achieve similar functionality with use of Linux-provided schedulers such as cron(8) or at(1), or user-supplied programs.

`/cx/ux resume rebuild`

This command allows you to resume the rebuild operation on the specified unit `/cx/ux`.

This feature is intended only for 7000/8000 series controllers. 9000 series controllers have an on-board scheduler where rebuild operations can be scheduled to take place at specified start and stop times. The `/cx/ux resume rebuild` function is provided to enable 7000/8000 users to achieve similar functionality with use of Linux-provided schedulers such as cron(8) or at(1), or user supplied programs.

`/cx/ux stop verify`

This command stops a background verification process on the specified unit `/cx/ux`. Table 7 on page 60 shows the supported matrix as a function of the controller model and logical unit type.

/cx/ux flush

This command allows you to flush the write cache on the specified unit **/ux** associated with controller **/cx**. Note that this command does not apply to spare unit types.

/cx/ux set autoverify=on|off

This feature only applies to 9000 series controllers.

This command allows you to turn on and off the autoverify operation on a specified unit **/cx/ux** during allocated schedule windows.

You can use the **show verify** command to display the existing schedule windows. By default, autoverify is **off**.

Auto-verify allows the controller to run the verify function once every 24 hours. If verify schedule windows are set up and enabled, then the controller will only start an automatic verify task during the schedule time slots. If the verify takes longer than the schedule window, the verify process will be paused and restarted during the next verify schedule window. For additional information, see “Setting Up a Verify Schedule” on page 46.

/cx/ux set cache=on|off [quiet]

This command allows you to turn **on** or **off** the write cache for a specified unit **/cx/ux**. This feature is supported on both 7000/8000 and 9000 models.

By default, cache is **on**.

Write cache includes the disk drive cache and controller cache. Note that for some configuration types, there is only disk drive cache and no controller cache (for example, JBOD).

The following table shows the supported RAID types for caching as a function of controller model and logical unit type. N/A (Not Applicable) refers to cases where the given logical unit type is not supported on a particular controller model.

Table 8: Supported RAID Types for Caching

Model	R0	R1	R5	R10	Single	Spare
7K/8K	Yes	Yes	Yes	Yes	N/A	No
9K	Yes	Yes	Yes	Yes	Yes	No

The **quiet** attribute turns off interactive mode.

`/cx/ux set identify=on|off`

This feature only applies to 9550SX and 9590SE model controllers.

This command allows you to identify a unit within an enclosure.

When that the EPCT (Enclosure Port Configuration Table) equates identify with blinking, as it does in the 3ware Sidecar Enclosure, the LEDs associated with the drive slots of the specified unit blink.

Example:

```
//localhost> /c0/u0 set identify=on  
Sending Identify request for unit /c0/u0 to [on] ... Done.
```

`/cx/ux set ignoreECC=on|off`

This feature only applies to 9000 series controllers.

This command allows you to set the ignoreECC policy for a given unit.

When ignoreECC policy is set to **off**, if a rebuild process encounters bad sectors on the source disk, the rebuild will fail. When ignoreECC is set to **on**, such errors are ignored, and the rebuild will continue. When you use ignoreECC, a file system check is recommended following the rebuild, to insure data integrity.

By default, ignoreECC is **off**.

`/cx/ux set name=string`

This command allows you to name the unit with an arbitrary name. You can use this name in conjunction with the unit serial number to cross-reference with the unit. The system does not check to ensure uniqueness of names, so be careful to assign different names to each unit.



Note: The unit's serial number is automatically assigned when the unit is created and is not changeable.

`/cx/ux set qpolicy=on|off`

This command applies only to 9550SX and 9590SE controllers.

This command sets the queue policy of the firmware. If the queue policy is **on**, the firmware utilizes the drive queuing policy. If some of the drives in the unit do not support any queuing policy, this policy will have no effect on those drives.

By default, `qpolicy` is off.

Example:

```
//localhost> /c0/u5 set qpolicy = on
Setting Command Queuing Policy for unit /c0/u5 to [on] ... Done.
```

`/cx/ux set storsave=protect|balance|perform [quiet]`

This command applies only to 9550SX and 9590SE model controllers.

This command sets the `storsave` policy to be either **protect**, **balance**, or **perform** when the unit write cache is enabled. The default setting is **protect**.

The `storsave` policy adjusts several factors that control the balance between protection and performance on a unit. There is a trade-off among the available settings. The following description about the settings should help you to decide which one is suitable to you and your application. You will find further discussion of this setting in the *3ware Serial ATA RAID Controller User Guide*, under “About StorSave Profile Levels” on page 48.

protect provides the maximum data protection among the controller settings. When `storsave` is set to **protect** mode, it means:

- When the unit becomes degraded, the write cache will be disabled.
- Write journaling is enabled. All data flushing from controller cache will be flushed to media.
- Incoming FUA (Force Unit Access) host requests will be honored unless a BBU is installed and enabled, in which case, they will be ignored.

perform provides the maximum performance and least data protection of the three controller settings. When `storsave` is set to **perform** mode, it means:

- When the unit becomes degraded, the write cache will not be disabled.
- Write journaling is disabled. All data flushing from controller cache will be flushed to disk. If a BBU is present, this essentially disables the BBU for this unit.
- Incoming FUA (Force Unit Access) host requests will be honored.

If you set the `storsave` policy to **perform**, a confirmation message will warn you that there could be data loss in the event of a power failure.

quiet is used for non-interactive mode. No confirmation is given and the command is executed immediately. This is useful for scripting purposes.

balance provides more data protection than `perform` mode but less data protection than `protect` mode, and provides better performance than `protect` mode but less performance than `perform` mode. When `storsave` is set to the **balance** mode, it means:

- When the unit becomes degraded, the write cache will not be disabled.

- Write journaling is disabled, if no BBU is present, and is enabled, if a BBU is present. All data flushing from controller cache will be flushed to media if a BBU is installed and enabled. Otherwise, data will be flushed to disk only.
- Incoming FUA (Force Unit Access) host requests will be honored unless a BBU is installed and enabled, in which case, they will be ignored.

For additional information, see “Setting the StorSave Profile for a Unit” in the *3ware Serial ATA RAID Controller User Guide*.

Example:

```
//localhost> /c0/u5 set storsave=protect
Setting Command Storsave Policy for unit /c0/u5 to [protect] ...
Done.
```

`/cx/ux migrate type=RaidType [disk=p:-p]
[group=3|4|5|6] [stripe=Stripe] [noscan] [nocache]
[autoverify]`

This feature only applies to 9000 series controllers.

This command allows you to change the existing configuration of a unit with **type=*RaidType***. You can make three types of changes:

- Increase the capacity
- Change the RAID level (with the same or increased capacity)
- Change the stripe size

The unit that results from the migration is subject to the same rules and policies that apply when creating a new unit with the **/cx add** command. For example, a valid number of disks and parameters must be specified.

The destination unit must use all source disks and potentially augment the number of disks in the **disk=*p:-p*** disk list. Unspecified parameters are assigned the default values (stripe size of 64K, write cache enabled, autoverify disabled, and ignoreECC disabled). Both source name and serial number will be carried over to the destination unit.

A special case of this command is when the source unit has a type of RAID1 and destination unit has a type of single. In this case, the migrate command splits both drives into two identical single disks. The disk name will be duplicated on the destination units, but the source unit serial number will not be carried over to the new unit. The new destination unit will have its own serial number.



Warning: Do not split a RAID 1 unit if you are booted from the drives or if the unit is mounted. Once the unit is split, any pending writes cannot be written to the second drive. In addition, the file system on the drive will not be clean.

type=RaidType specifies the RAID type of the destination unit. Possible unit types include **raid0**, **raid1**, **raid5**, **raid10**, **raid50**, or **single**.

For example, **type=raid5** indicates the destination unit is RAID-5. The **type=single** is a special case of the migrate command. It splits the source unit RAID-1 or TWINSTOR into multiple Single units.



Note: You can only migrate a unit to a RAID level that has the same or more capacity as the existing one. A four-drive RAID 5 unit can migrate to a four-drive RAID 0, but a four-drive RAID 0 unit cannot migrate to a four-drive RAID 5, without adding another drive, due to the need for additional storage capacity for parity bits.

The following table illustrates valid migration paths:

Table 9: Valid Migration Paths

Source	Destination					
	R0	R1	R5	R10	Single	Spare
R0	Yes	No	Yes	Yes	No	No
R1	Yes	No	Yes	Yes	Yes	No
R5	Yes	No	Yes	Yes	No	No
R10	Yes	No	Yes	Yes ^a	No	No
Single	Yes	Yes	Yes	Yes	No	No
Spare	No	No	No	No	No	No

a. When migrating a RAID 10 to a RAID 10, the only change you can make is the stripe size.

disk=p:p.. consists of a list of ports (disks) to be used in addition to the source disks in the construction of the destination unit. One or more ports can be specified. Multiple ports can be specified using a colon (:) or a dash (-) as port index separators. A dash indicates a range and can be mixed with colons. For example **disk=0:1:2-5:9:12** indicates port 0, 1, 2 through 5 (inclusive), 9 and 12.

group=3/4/5/6 indicates the number of disks per group for a RAID 50 type. (This attribute can only be used when `type=raid50`.) Recall that a RAID 50 is a multi-tier array. At the bottom-most layer, N number of disks per group are used to form the RAID 5 layer. These RAID 5 arrays are then integrated into a RAID 0. This attribute allows you to specify the number of disks in the RAID 5 level. Valid values are 3, 4, 5 and 6. For example **group=3** indicates 3 disks of RAID 5 at the bottom layer of RAID 50.

Note that a sufficient number of disks are required for a given pattern or disk group. For example, given 6 disks, specifying 3 will create two RAID 5 arrays. With 12 disks, specifying 3 will create four RAID 5 arrays under the RAID 0 level. With only 6 disks a grouping of 6 is not allowed, as you would basically be creating a RAID 5.

The default RAID 50 grouping varies, based on number of disks. For 6 and 9 disks, default grouping is 3. For 8 disks, the default grouping is 4. For 10 disks, the default grouping is 5, and for 12 disks, the default grouping is 4. In the case of 12, the disks could be grouped into groups of 3, 4, or 6 drives. A grouping of 4 is set by default as it provides the best of net capacity and performance.

Note that RAID-10 always has **group=2**, so an attribute specifying it's group is not necessary.

stripe=Stripe consists of the stripe size to be used. The following table illustrates the supported and applicable stripes on unit types and controller models. Stripe size units are in KB (kilobytes).

Table 10: Supported Stripe Sizes

Model	R0	R1	R5	R10	Spare	Single
9K	16	N/A	16	16	N/A	N/A
	64		64	64		
	256		256	256		

noscan attribute instructs CLI not to notify the operating system of the creation of the new unit. By default CLI will inform the operating system. One application of this feature is to prevent the operating system from creating block special devices such as `/dev/sdb` and `/dev/sdc` as some implementations might create naming fragmentation and a moving target.

nocache attribute instructs CLI to disable the write cache on the migrated unit. Enabling write cache increases write performance at the cost of potential data loss in case of sudden power loss (unless a BBU or UPS is installed). By default the cache is enabled. To avoid the possibility of data loss in the event of a sudden power loss, it is recommended not to set `nocache` unless there is a BBU (battery backup unit) or UPS (uninterruptable power supply) installed..

autoverify attribute enables the autoverify attribute on the unit that is to be migrated. For more details on this feature, see “/cx/ux set autoverify=on|off” on page 61.

Migration Process

In all cases of migration, the background migration process must be completed before the newly sized unit is available for use. You can continue using the original unit during this time. Once the migration is finished, a reboot will be required if you are booted from the unit. For secondary storage, depending on your operating system, you may need to first unmount the unit, then use CLI commands to ‘remove’ and ‘rescan’ the unit so that the operating system can see the new capacity, and then remount the unit. For details see “/cx/ux remove [noscan] [quiet]” on page 58 and “/cx rescan [noscan]” on page 38.

You may also need to resize the file system or add a new partition. For instructions, consult the documentation for your operating system.



Warning: It is important that you allow migration to complete before adding drives to the unit. Making physical changes to the unit during migration may cause the migration process to stop, and can jeopardize the safety of your data.

Example of splitting a mirror

```
//localhost> /c1/u3 migrate type=single
```

Indicates that u3 is a TWINSTOR or RAID-1 and the Migrate command splits u3 to u3 and ux with a RAID type of Single.

Example of capacity expansion

```
//localhost> /c0/u0 migrate type=raid10 disk=3-4 stripe=16
```

Indicates that the destination unit has a RAID type of raid10 and has the disk 3 and 4 in addition to all the disks in the existing unit u0.

Example of migrate output

The following is an example of how migrating units will be displayed. In this example, the report indicates that /c0/u0 is a migrating unit with 67% completion. The report also indicate that Source Unit su0 is of type RAID-1 and Destination Unit du0 is of type RAID-10.

```
//localhost> /c0/u0 migrate type=raid10 disk=2-3 stripe=16
Sending migration message to /c0/u0 ... Done.
```

```
3ware CLI> /c0 show
```

Unit	UnitType	Status	%Cmpl	Stripe	Size(GB)	Cache	AVerify	IgnECC
u0	Migrator	MIGRATING	67%	-	74.4951	ON	OFF	OFF

Port	Status	Unit	Size	Blocks	Serial
p0	OK	u0	74.53 GB	156301488	3JV2Q1VA
p1	OK	u0	232.83 GB	488281250	WD-WMAEH15764
p2	OK	u0	232.83 GB	488281250	WD-WMAEH17004
p3	OK	u0	232.83 GB	488281250	WD-WMAEH17000
p4	NOT-PRESENT	-	-	-	-
p5	NOT-PRESENT	-	-	-	-
p6	NOT-PRESENT	-	-	-	-
p7	NOT-PRESENT	-	-	-	-
p8	NOT-PRESENT	-	-	-	-
p9	NOT-PRESENT	-	-	-	-
p10	NOT-PRESENT	-	-	-	-
p11	NOT-PRESENT	-	-	-	-

```
3ware CLI> /c0/u0 show
```

Unit	UnitType	Status	%Cmpl	Port	Stripe	Size(GB)	Blocks
u0	Migrator	MIGRATING	67%	-	-	-	-
su0	RAID-1	OK	-	-	-	74.4951	156227584
su0-0	DISK	OK	-	p0	-	74.4951	156227584
su0-1	DISK	OK	-	p1	-	74.4951	156227584
du0	RAID-10	OK	-	-	16K	148.99	312455168
du0-0	RAID-1	OK	-	-	-	-	-
du0-0-0	DISK	OK	-	p0	-	74.4951	156227584
du0-0-1	DISK	OK	-	p1	-	74.4951	156227584
du0-1	RAID-1	OK	-	-	-	-	-
du0-1-0	DISK	OK	-	p2	-	74.4951	156227584
du0-1-1	DISK	OK	-	p3	-	74.4951	156227584

Port Object Commands

Port Object Messages are commands that provide information and perform actions related to a specific disk, attached to a port, such as **/c0/p0**. You use port object commands for such tasks as seeing the status, model, or serial number of the drive.

Syntax

```
/cx/px show
/cx/px show attribute [attribute ...] where attributes are:
    capacity|firmware|identify (9550SX and 9590SE only)|
    lspeed (9550SX and 9590SE only)|model|
    ncq (9550SX and 9590SE only)|serial|smart|status
/cx/px show all

/cx/px remove [noscan][quiet]
/cx/px set identify=on|off (9550SX and 9590SE only)
```

/cx/px show

This command shows summary information about the specified disk attached to port **/cx/px**. Typical information looks like:

Example:

```
//localhost> /c1/p5 show
```

Port	Status	Unit	Size	Blocks	Serial
p5	OK	u0	149.05 GB	312581808	3JS0L9QW

The above report indicates that port 5 of controller 1 is attached to one 1 disk with status OK participating in unit 0.

/cx/px show *attribute [attribute ...]*

This command shows the current setting of the given attributes on the specified port or drive. One or many attributes can be requested. Specifying an invalid attribute will terminate the loop. Possible attributes are: capacity, firmware, identify (9550SX and 9590SE only), lspeed (9550SX and 9590SE only), model, ncq (9550SX and 9590SE only), serial, smart, and status.

/cx/px show status

This command displays the status of the drive attached to the specified port. (Definitions of the drive statuses are available in the *3ware Serial ATA RAID Controller User Guide*.)

Example:

```
//localhost> /c0/p5 show status
/c0/p5 Status = OK
```

/cx/px show model

This command displays the model of the drive attached to the specified port.

Example:

```
//localhost> /c0/p5 show model
/c0/p5 Model = WDC WD1600BB-00DAA0
```

/cx/px show serial

This command displays the serial number of the drive attached to the specified port.

Example:

```
//localhost> /c0/p5 show serial
/c0/p5 Serial = WD-WMACK140649
```

/cx/px show firmware

This command displays the firmware version of the drive attached to the specified port.

Example:

```
//localhost> /c0/p5 show firmware
/c0/p5 Firmware Version = 65.13G65
```

/cx/px show identify

This command applies only to 9550SX and 9590SE model controllers that have chassis control hardware (enclosure services) attached.

This command shows whether the LED of the drive attached to the specified port is set to **on** or **off**. For details, see “/cx/px set identify=on|off” on page 73.

Example:

```
//localhost> /c0/p5 show identify
/c0/p5 Identify Status = on
```

/cx/px show ncq

This command applies only to 9550SX and 9590SE model controllers.

This command displays the NCQ (Native Command Queuing) information for the drive attached to the specified port, including whether NCQ is supported by the drive, and whether it is enabled at the drive.

For queuing to be used, it must be enabled for the unit and supported by the drive.

Example:

```
//localhost> /c0/p5 show ncq
/c0/p5 NCQ Supported = No
/c0/p5 NCQ Enabled = No
```

/cx/px show lspeed

This command applies only to 9550SX and 9590SE model controllers.

This command displays the maximum SATA link speed supported by the drive attached to the port and the present SATA link speed setting.

Example:

```
//localhost> /c0/p5 show lspeed
/c0/p5 SATA Link Speed Supported = 3.0 Gb/s
/c0/p5 SATA Link Speed = 3.0 Gb/s
```

/cx/px show capacity

This command displays the capacity of the drive attached to the specified port in two formats—GB and blocks. Note that of this version, the GB format is computed based on division by 1000 (not 1024).

Example:

```
//localhost> /c0/p5 show capacity
149.05 GB (312581808)
```

/cx/px show smart

This command extracts SMART (Self Monitoring Analysis and Reporting) data from the specified disk. Because the data is extracted live from the disk, this command can be used to get the most recent data about the presence or absence of a disk.

The SMART data is displayed in hexadecimal form.

Example:

```
//localhost> /c0/p5 show smart
/c0/p5 Drive SMART Data:
10 00 01 0B 00 C8 C8 00 00 00 00 00 00 03 07
00 9A 96 BC 14 00 00 00 00 04 32 00 64 64 7A
00 00 00 00 00 00 05 33 00 C8 C8 00 00 00 00 00
...
00 00 00 00 00 00 00 00 00 00 00 00 00 00 2C
```



Note: The SMART data is not decoded. If the drive attached to the specified port is not present or if there are cabling problems reaching the drive, CLI will return an error. This can be one way of detecting whether or not a drive is present.

/cx/p x show all

This command shows the current setting for all port-related attributes: status, model, serial, firmware, capacity, and smart.

Example:

```
//localhost> /c0/p0 show all
/c0/p0 Status = OK
/c0/p0 Model = Maxtor 7B300S0
/c0/p0 Firmware Version = BANC1980
/c0/p0 Serial = B605X31H
/c0/p0 Capacity = 279.48 GB (586114704 Blocks)
/c0/p0 Identify Status - NA
/c0/p0 SATA Link Speed Supported = 1.5 Gb/s
/c0/p0 SATA Link Speed = 1.5 Gb/s
/c0/p0 NCQ Supported = No
/c0/p0 NCQ Enabled = No
/c0/p0 Belongs to Unit = u1

/c0/p0 Drive Smart Data:
0A 00 01 0F 00 3D 33 25 8C BA 03 00 00 00 03 03
00 61 60 00 00 00 00 00 00 00 04 32 00 64 64 00
00 00 00 00 00 00 05 33 00 64 64 00 00 00 00 00
00 00 07 0F 00 4E 3E 05 13 D8 03 00 00 00 09 32
...
00 00 00 00 00 00 00 00 00 00 00 00 00 00 8D
```

/cx/px remove [noscan] [quiet]

This command allows you to remove (or export) a port (or drive) **/cx/px**. Exporting a port instructs the firmware to remove the specified port from its pool of managed ports, but does not retain the DCB (Disk Configuration Block) metadata on the attached disk. You can import (or re-introduce) the port by rescanning the controller.

noscan is used to not inform the operating system of this change. The default is to inform the operating system.

quiet is for non-interactive mode.



Warning: Use caution when using this command as this operation will degrade any redundant units. This command will fail if you attempt to remove a drive from a non-redundant unit. After the drive is removed in CLI it can be removed physically, without powering down the system if a hot swap carrier is available. System hangs and damage can occur if a hot swap carrier is not used.

/cx/px set identify=on|off

This command applies only to 9550SX and 9590SE model controllers.

This command sets the LED status of the port to **on** or **off**. If identify is set to **on**, the firmware activates the setting of the corresponding LED of the port on the controller and causes it to blink.

Note: This command is equivalent to “/ex/slotx set identify=on|off” on page 82.



Note: This feature requires a chassis with support for Chassis Control Unit (CCU), such as the 3ware Sidecar, or a chassis from a third-party vendor. The CCU may also support the monitoring of fan speed, temperature sensors, door lock, and so forth, depending on the level of chassis vendor support. Check the 3ware support web site for future updates on a list of chassis vendors that support the CCU, as they become available. For additional information about enclosure-related commands, see “Enclosure Object Commands” on page 79.

Example:

```
//localhost> /c0/p5 set identify=on
Setting Port Identify on /c0/p5 to [on] ... Done.
```

BBU Object Commands

BBU (Battery Backup Unit) Object Commands are commands that provide information and perform actions related to a specific BBU installed on a specific controller, such as `/c0/bbu`.

This object is only available on 9000 series controllers on which a BBU is actually installed. (The BBU is not supported on 9590SE-4ME.)

Syntax

```

/cx/bbu show (9000 only)
/cx/bbu show attribute [attribute ...] where attributes are:
    batinst|bootloader|cap|fw|lasttest|pcb|ready|serial|
    status|temp|volt
/cx/bbu show all (9000 only)

/cx/bbu test [quiet] (9000 only)
    Warning: May take up to 24 hours to complete. Write cache
    will be disabled during the test.
/cx/bbu enable (9000 only)
/cx/bbu disable [quiet] (9000 only)

```

`/cx/bbu show`

This command presents a summary report on the specified BBU object.

Example:

```

//localhost> /c0/bbu show
Name OnlineState BBUReady Status Volt Temp Hours LastCapTest
-----
bbu ON No Testing OK OK 72 01-Jul-2004

```

The command output indicates that the battery capacity was last measured on 01-Jul-2004. The battery is estimated to last for 72 hours from the last tested date. In this example, the BBU unit is currently testing the battery. Both voltage and temperature are normal. The BBU is not ready to backup the write cache on the controller (due to the testing). (For complete information about the BBU, see the user guide that came with your 3ware RAID controller).



Note: If the BBU is either not present or disabled, the following will be displayed after the command `//localhost> /c0/bbu show`.

Error: (CLI:053) Battery Backup Unit is not present.

`/cx/bbu show attribute [attribute ...]`

This command shows the current setting of the given attribute(s) on the BBU board. One or many attributes can be specified. Specifying an invalid attribute will terminate the loop. Possible attributes are: batinst, bootloader, cap, fw, lasttest, pcb, ready, status, serial, temp, volt.

`/cx/bbu show status`

This command shows the status of the BBU. Possible values are:

Testing. A battery test is currently in progress. This test may take up to 24 hours to complete. During the test, the BBU is **not** capable of backup operation and the write cache of the RAID controller is also disabled. If the test is completed with no error and the BBU status changes to WeakBat or OK, the write cache will be re-enabled. If a Fault, Failed or Error occurs during the test, the write cache remains in the disabled state until the problem is fixed.

Charging. The BBU is currently charging the battery. Charging is started automatically by the BBU whenever necessary. During charging, the BBU is not capable of backup operation and the write cache is disabled. Once the test is completed with no error and the BBU status changes to OK, the write cache will be re-enabled. If a FAULT or ERROR occurs during the test, the write cache remains in the disabled state until the problem is fixed.

Fault. A battery fault is detected. The BBU is not capable of backup operation and the write cache is disabled. Replace the battery and/or the BBU board as soon as possible so that the write cache will be enabled again.

Error. A BBU error is detected. The BBU is not capable of backup operation and the write cache is disabled. Replace the battery and/or the Battery Backup Unit as soon as possible so that the write cache will be enabled again.

Failed. The battery failed a test. In this state, the BBU is not capable of backup operation and the write cache is disabled. We recommend you replace the battery and/or the Battery Backup Unit as soon as possible so that the write cache will be enabled again.

WeakBat. The BBU is functioning normally and is online and capable of backing up the write cache. However, the battery is weak and should be replaced.

OK. The BBU is ready, online and capable of backing up the write cache.

- (dash) A battery is not present or a Battery Backup Unit is not installed

/cx/bbu show batinst

This command **shows** the date when the current battery was installed.

/cx/bbu show lasttest

This command **shows** the date the battery capacity was last measured. If the battery capacity test has never been run, then 'xx-xxx-xxxx' will be displayed.



Note: The estimated BBU capacity hours displayed is based on the measurement taken during the last test. If you have not run the BBU test command for some time, this number can be misleading. For information about running a test, see “/cx/bbu test [quiet]” on page 77.

/cx/bbu show volt

This command shows the voltage status of the battery. The status can be OK, HIGH, LOW, TOO-HIGH, and TOO-LOW. The HIGH and LOW are in warning range. TOO-HIGH and TOO-LOW are out of the operating range and indicate that it is time to replace the battery. (Contact AMCC to obtain a replacement battery.)

/cx/bbu show temp

This command **shows** the temperature status of the battery. The status can be OK, HIGH, LOW, TOO-HIGH, and TOO-LOW. The HIGH and LOW are in warning range. TOO-HIGH and TOO-LOW are out of the operating range and indicate that it may be time to replace the battery. (Contact AMCC to obtain a replacement battery.)

/cx/bbu show cap

This command **shows** the battery capacity in hours.

A value of '0 hours' will be displayed if the battery capacity test has never been run.



Note: The estimated BBU capacity hours displayed is based on the measurement taken during the last test. If you have not run the BBU test command for some time, this number can be misleading. You can use the command **/cx/bbu show lasttest** to check the date of the last test. For information about running a test, see “/cx/bbu test [quiet]” on page 77.

/cx/bbu show serial

This command **shows** the BBU serial number.

/cx/bbu show fw

This command **shows** the BBU firmware version number.

/cx/bbu show pcb

This command **shows** the PCB revision number on the BBU.

/cx/bbu show bootloader

This command **shows** the BBU's boot loader version.

/cx/bbu show all

This command shows the current settings of all BBU-related attributes: ready, status, batinst, lasttest, volt, temp, cap, serial, fw, pcb, bootloader.

Example:

```
//localhost> /c1/bbu show all
/c1/bbu Firmware Version          = BBU: 1.04.00.007
/c1/bbu Serial Number             = Engineering Sample.
/c1/bbu BBU Ready                 = Yes
/c1/bbu BBU Status                = OK
/c1/bbu Battery Voltage           = OK
/c1/bbu Battery Temperature       = OK
/c1/bbu Estimated Backup Capacity = 241 Hours
/c1/bbu Last Capacity Test        = 22-Jun-2004
/c1/bbu Battery Installation Date = 20-Jun-2004
/c1/bbu Bootloader Version        = BBU 0.02.00.002
/c1/bbu PCB Revision              = 65
//localhost>
```

/cx/bbu test [quiet]

This command starts the battery capacity test. The test may take up to 24 hours to complete. During the test, the BBU is not capable of backup operation and the write cache of all units attached to that controller is disabled. Once the test is completed with no error and the BBU status returns to OK, the write cache will be re-enabled.



Note: Once started, the test can not be terminated before it completes. Write cache cannot be enabled until the test completes.

AEN (Asynchronous Event Notification) messages are also generated by controllers to notify the user of the command status.

Check for AENs with the alarms command **/cx show alarms [reverse]**. Using the “reverse” attribute displays the most recent AEN message at the bottom of the list. (For a list of all AENs, see the user guide that came with your 3ware RAID controller.)

`/cx/bbu enable`

This command enables BBU detection on the controller. If the BBU is Ready, the controller will utilize BBU functionality in the event of a power failure.

`/cx/bbu disable [quiet]`

This command disables BBU detection on the controller. When disabled, the controller ignores the existence of the BBU and will show no BBU is installed even if a BBU is physically attached.

Enclosure Object Commands

Enclosure object commands provide information and perform actions related to a specific enclosure, such as **/e0** and its elements, such as **/e0/slot0**.

Enclosure object elements include slot, fan, and temperature sensor elements.

These commands support the 9550SX and 9590SE controllers. If you purchased an enclosure directly from a third-party vendor, the appropriate EPCT (Enclosure Port Configuration Table) must have been downloaded to the controller in order to take advantage of these commands.

If you have a 3ware Sidecar Kit for use with a Power Mac G5, the required EPCT file is already installed.

Syntax

```

/ex show                                     (9550SX and 9590SE only)
/ex show attribute [attribute ...] where attributes are:
      controllers|slots|fans|temp
/ex show all                                 (9550SX and 9590SE only)

/ex/slotx show                               (9550SX and 9590SE only)
/ex/slotx show identify                     (9550SX and 9590SE only)
/ex/slotx set identify=on|off               (9550SX and 9590SE only)

/ex/fanx show                               (9550SX and 9590SE only)
/ex/tempx show                              (9550SX and 9590SE only)

```

/ex show

This command shows summary information on the specified enclosure **/ex**. This report consists of four parts; the **Enclosure** summary listing the present elements, a **Fan** summary section listing of all present fans, a **Temperature Sensor** summary section listing of all present temperature sensors and a **Slot** summary section listing of slots and associated information for the specified enclosure.

Typical output looks like:

```

//localhost> /e0 show

Encl          Controllers
-----
e0            /c0

Fan           Status
-----
fan0         OK

```

TempSensor	Temperature
temp0	24~C(75~F)

Slot	Status	Port	Identify
slot0	OK	/c0/p0	No
slot1	OK	/c0/p1	Yes
slot2	NO-DEVICE	-	No
slot3	NO-DEVICE	-	No

/ex show attribute [attribute ...]

This command shows the current setting of the given attribute(s). One or many attributes can be requested. An invalid attribute will terminate the loop. Possible attributes are: controllers, protocol, slots, fans, and temp.

/ex show controllers

This command lists the controller associated with enclosure */ex*.

Example:

```
//localhost> /e0 show controllers  
/e0 Connected to /c0 controller.
```

/ex show slots

This command reports the slots in enclosure */ex* and their associated information.

Example:

```
//localhost> /e0 show slots
```

Slot	Status	Port	Identify
slot0	OK	/c0/p0	No
slot1	OK	/c0/p1	Yes
slot2	NO-DEVICE	-	No
slot3	NO-DEVICE	-	No

/ex show fans

This command lists the fans in enclosure **/ex** and shows their status. Possible statuses are OK and Unknown.

Example:

```
//localhost> /e0 show fans
```

Fan	Status
fan0	OK

/ex show temp

This command lists the temperature sensors in enclosure **/ex** and the current temperature.

The maximum temperature for successful use of a drive should be noted in the documentation for the drive.

Example:

```
//localhost> /e0 show temp
```

TempSensor	Temperature
temp0	24~C(75~F)

/ex show all

This command shows the current settings of all attributes for enclosure **/ex**.

/ex/slotx show

This command shows information about the specified **/slotx** on the specified enclosure **/ex**. The slot name is followed by its status. If a slot has been inserted with a drive and no fault has been detected, the status is OK. If the slot is empty the status would indicate NO-DEVICE. The port that is correlated to the slot is indicated in the next column. If no device is found in the slot, that is indicated with a dash (-) in the Port column. The final column shows whether “identify” is currently set for the specified slot.

Example:

```
//localhost> /e0/slot1 show
```

Slot	Status	Port	Identify
slot1	OK	/c0/p1	Yes

/ex/slotx show identify

This command shows the identify status of the specified slot. The status can be either **on** or **off**.

Example:

```
//localhost> /e0/slot1 show identify
/e0/slot1 Identify status = on
```

/ex/slotx set identify=on|off

This command causes the slot to be identified by blinking the LED associated with it, or turns off identification of the LED for this slot.

Setting identify to **on** will cause the LED associated with that slot to blink, provided that the EPCT has been set to associate “identify” with “blinking,” as is the case in the 3ware Sidecar.

Note: This command is equivalent to “/cx/px set identify=on|off” on page 73.

Example:

```
//localhost> /e0/slot1 set identify=on
Sending Identify request to Drive Slot /e0/slot0 to [on] ...
Done.
```

/ex/fanx show

This command shows the information about the specified fan element **/fanx** in the specified enclosure **/ex**. The fan name is followed by its status. If a fan is on and no fault has been detected, the status would indicate OK.

Example:

```
//localhost> /e0/fan0 show
Fan          Status
-----
fan0        OK
```

/ex/tempx show

This command shows the information about the specified temperature sensor element **/fanx** in the specified enclosure **/ex**. The temperature sensor name is followed by the temperature sensed in the enclosure unit.

Example:

```
//localhost> /e0/temp0 show
TempSensor   Temperature
-----
temp0        24~C(75~F)
```

Help Commands

The Help commands provides brief on-line help.

You can get overview help by typing Help at the top-level prompt. This displays a brief definition of commands in both the new syntax and the legacy syntax. (For an example, see the discussion of the command “help” on page 85.)

You can also get help with specific commands, by entering *help* before an object name, or by typing a question mark (?) at the point in a command where you are uncertain what the attributes are.

Help with specific commands

If you enter the help command at the top level, you are considered to be in the Shell Object, and the help command will provide help on the Shell commands focus, show, flush, rescan, and commit. Using the help command on objects (such as */cx*, */cx/ux*, */cx/px*, */cx/bbu*, */ex*, */ex/slotx*, */ex/fanx*, and */ex/tempx*), displays all possible sub-commands associated with the object.

For example: help on the controller object */cx*, will display all the sub-commands associated with the controller */cx*, like this:

```
//localhost> help /cx
/cx show
/cx show attribute [attribute ...]           where attribute is:
achip|allunitstatus|autocarve|autorebuild(9550SX and 9590SE
only)|bios|carvesize(9000 series)|ctlbus(9550SX and 9590SE
only)|driver|drivestatus|exportjbod|firmware|memory|
model|monitor|numdrives|numports|numunits|pcb|pchip|
serial|spinup|stagger|unitstatus|ondegrade(9000S only)
/cx show all      where all means attributes and configurations.
/cx show diag
/cx show alarms [reverse]
/cx show rebuild                                     (9000 only)
/cx show verify                                     (9000 only)
/cx show selftest                                    (9000 only)

/cx add type=<RaidType> disk=<p:-p..> [stripe=<Stripe>]
[noscan] [nocache][group=<3|4|5|6>] [autoverify] [ignoreECC]
[name=string (9000 only)] [storsave=<protect|balance|
perform[quiet]>(9550SX and 9590SE only)] RaidType={raid0,
raid1, raid5, raid10, raid50, single, spare, JBOD
(7000/8000 only)}
/cx add rebuild=ddd:hh:duration                     (9000 only)
/cx add verify=ddd:hh:duration                       (9000 only)
/cx add selftest=ddd:hh                             (9000 only)

/cx del rebuild=slot_id                             (9000 only)
/cx del verify=slot_id                              (9000 only)
```

```

/cx del selftest=slot_id (9000 only)

/cx set exportjbod=on|off (9000 only)
/cx set ondegrade=cacheoff|follow (9500S only)
/cx set spinup=nn (9000 only)
/cx set stagger=nn (9000 only)
/cx set autocarve=on|off (9000 only)
/cx set rebuild=enable|disable|<1..5> enable|disable
    for 9000 only)
/cx set verify=enable|disable|<1..5> (enable|disable
    for 9000 only)
/cx set selftest=enable|disable [task=UDMA|SMART](9000 only)
/cx set autorebuild=on|off (9550SX and 9590SE only)

/cx update fw=filename_with_path [force] (9000 only)
/cx flush
/cx commit (Windows only) (Also known as shutdown)
/cx start mediascan (7000/8000 only)
/cx stop mediascan (7000/8000 only)
/cx rescan [noscan] NOTE: Does not import non-JBOD on
    7000/8000 models.
//localhost>

```

Help with attributes

As you work with specific objects or commands, you can also use `?` to get help.

For example: If you enter the command `/c0 show` and then need help on what specific attribute syntax is possible, you can use `?` to get help as following:

```

//localhost> /c0 show ?

/cx show
/cx show attribute [attribute ...] where attribute is:
    achip|allunitstatus|autocarve(9000 series)|
    autorebuild(9550SX only)|bios|carvesize(9000series)|driver|
    drivestatus|exportjbod|firmware|memory|model|monitor|
    numdrives|numports|numunits|ctlbus(9550SX and 9590SE only)
    serial|ondegrade (9000S only)|pcb|pchip|spinup|stagger|
    unitstatus|
/cx show all where all means attributes and configurations.
/cx show diag
/cx show alarms [reverse]
/cx show rebuild (9000 only)
/cx show verify (9000 only)
/cx show selftest (9000 only)
//localhost>

```

help

This help command provide a table of contents, providing help with the overall navigation of the CLI commands. Typical output looks like the following.

```
//localhost> help
Copyright(c) 2004, 2005 Applied Micro Circuits Corporation
(AMCC). All rights reserved.

AMCC/3ware CLI (version 2.x)

Commands  Description
-----
show      Displays information about controller(s), unit(s) and port(s).
flush     Flush write cache data to units in the system.
rescan    Rescan all empty ports for new unit(s) and disk(s).
update    Update controller firmware from an image file
/cx       Controller specific commands.
/cx/ux    Unit specific commands.
/cx/px    Port specific commands.
/cx/bbu   BBU specific commands.                (9000 only)
/ex       Enclosure specific commands.           (9550SX and 9590SE only)
/ex/slotx Slot specific commands.
/ex/fanx  Fan specific commands.
/ex/tempx Enclosure Temperature Sensor specific commands.

Certain commands are qualified with constraints of controller type/model
support. Please consult the tw_cli documentation for explanation of the
controller-qualifiers.

The controller-qualifiers of the Enclosure commands (/ex) also apply to
Enclosure Element specific commands (e.g., /ex/elementx).

Type help <command> to get more details about a particular command. For more
detail information see tw_cli's documentation.
```

help show

This command provides specific show-related help, illustrating various ways to use the show command. It provides reports on Controllers, Units and Drives. See the section “Shell Object Commands” on page 19 for more information.

help flush

This command provides specific flush-related help, illustrating various ways to use the flush command. See the section “Shell Object Commands” on page 19 for more information.

help rescan

This command provides specific rescan related help, illustrating various ways to use the rescan command. See the section “Shell Object Commands” on page 19 for more information.

help update

This command provides specific update-related help. See “Shell Object Commands” on page 19 for more information.

help commit

This command provides specific commit related help, illustrating various ways to use the commit command. See the section “Shell Object Commands” on page 19 for more information.

help focus

This command provides specific focus related help, illustrating various ways to use the focus command. See the section “Shell Object Commands” on page 19 for more information.

help /cx

This command provides specific controller **/cx** related help, illustrating various commands associated with the controller **/cx**. See the section “Controller Object Commands” on page 25 for more information.

help /cx/ux

This command provides specific unit **/cx/ux** related help, illustrating various commands to use on a unit **/cx/ux**. See the section “Unit Object Commands” on page 53 for more information.

help /cx/px

This command provides specific **/cx/px** related help, illustrating various ways to use the **/cx/px** command. See the section “Port Object Commands” on page 69 for more information.

help /cx/bbu

This command provides specific **/cx/bbu** related help, illustrating various ways to use the **/cx/bbu** command. See the section “BBU Object Commands” on page 74 for more information.

help /ex

This command provides specific enclosure **/ex** related help, illustrating various commands associated with the enclosure **/ex**. See the section “Enclosure Object Commands” on page 79 for more information.

help /ex/slotx

This command provides specific slot **/ex/slotx** related help, illustrating various ways to use **/ex/slotx**. See the section “Enclosure Object Commands” on page 79 for more information.

help /ex/fanx

This command provides specific fan **/ex/fanx** related help, illustrating various ways to use the **/ex/fanx** command. See the section “Enclosure Object Commands” on page 79 for more information.

help /ex/tempx

This command provides specific temperature sensor **/ex/tempx** related help, illustrating various ways to use the **/ex/tempx** command. See the section “Enclosure Object Commands” on page 79 for more information.

Return Code

While informative messages are written to standard output, error messages are written to standard error. On success, 0 is returned. On failure, 1 is returned.